

**PENGEMBANGAN SISTEM INFORMASI DATA PASIEN
SEKSI REHABILITASI BNN KOTA MALANG MENGGUNAKAN
METODE *ITERATIVE INCREMENTAL***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Stefanus Eka Prastya
NIM: 145150401111018



PROGRAM STUDI SISTEM INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI DATA PASIEN SEKSI REHABILITASI BNN
KOTA MALANG MENGGUNAKAN METODE *ITERATIVE INCREMENTAL*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Stefanus Eka Prastya
NIM: 145150401111018

Skripsi ini telah diuji dan dinyatakan lulus pada
25 Juli 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Mochamad Chandra Saputra, S.Kom., M.Eng.

NIK: 201609 860106 1 001

Dosen Pembimbing II



Dioko Pramono, S.T., M.Kom.

NIP: 19780108 200501 1 002

Mengetahui

Ketua Jurusan Sistem Informasi



Herman Tolle, Dr. Eng., S.T, M.T

NIP: 19740823 200012 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis di sitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 25 Juli 2018



Stefanus Eka Prastya

NIM: 145150401111018

KATA PENGANTAR

Segala puji bagi Tuhan YME atas limpahan rahmat yang diberikan sehingga penulis dapat menyelesaikan skripsi berjudul “Pengembangan Sistem Informasi Data Pasien Seksi Rehabilitasi BNN Kota Malang Menggunakan Metode *Iterative Incremental*”. Dari aspek akademis, penelitian ini mengajarkan banyak ilmu yang belum saya dapatkan di kelas perkuliahan. Selain itu, penelitian ini juga mengajarkan bagaimana penulis dapat berkomunikasi dengan banyak orang, menyamakan persepsi yang saya miliki dengan pemangku kepentingan dan dosen pembimbing. Proses penelitian yang dilakukan juga mengajarkan untuk menjadi pribadi yang lebih sabar, menurunkan ego, dan senantiasa berbagi hal-hal yang bermanfaat yang berkaitan dengan kebutuhan pembelajaran. Penelitian ini tidak dapat diselesaikan tanpa dukungan dan bantuan dari orang-orang terdekat, sehingga saya ingin menyampaikan rasa terima kasih kepada:

1. Bapak Wayan Firdaus Mahmudy , S.Si, M.T, Ph.D. selaku Dekan Fakultas Ilmu Komputer Brawijaya
2. Bapak Herman Tolle , Dr. Eng., S.T, M.T selaku Kepala Jurusan Sistem Informasi Brawijaya
3. Bapak Suprpto, S.T., M.T. selaku Ketua Program Studi Sistem Informasi Brawijaya.
4. Bapak Mochamad Chandra Saputra , S.Kom., M.Eng, selaku pembimbing satu yang telah memberikan kesempatan untuk bergabung dalam kelompok riset dan memberikan topik penelitian, serta menjadi pembimbing yang sabar, solutif, dan memberikan contoh sebagai pribadi yang profesional dalam memberikan bimbingan kepada mahasiswa.
5. Bapak Djoko Pramono, S.T., M.Kom, selaku pembimbing dua yang mengajarkan ketegasan dan kedisiplinan dalam menyelesaikan kewajiban. Memberikan contoh sebagai pribadi yang sabar, solutif, dan profesional dalam memberikan bimbingan kepada mahasiswa.
6. Bapak Aditya Rachmadi, S.ST., M.TI selaku dosen penasihat akademik, segenap Bapak dan Ibu dosen serta karyawan yang telah mendidik dan membantu penulis selama menempuh pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya.
7. Ibu saya, Riwayati yang selalu memberikan dukungan doa, moril, materiil, mengajarkan ketegasan, kedisiplinan dalam menyelesaikan tugas, dan senantiasa memberikan contoh yang baik untuk menjadi pribadi yang sabar dan pekerja keras. Ayah saya, Srimulyono yang sangat saya yakini senantiasa memberikan dukungan doa dan senantiasa menjadi pribadi yang bersahabat dengan saya. Serta adik saya yang saya sayangi.

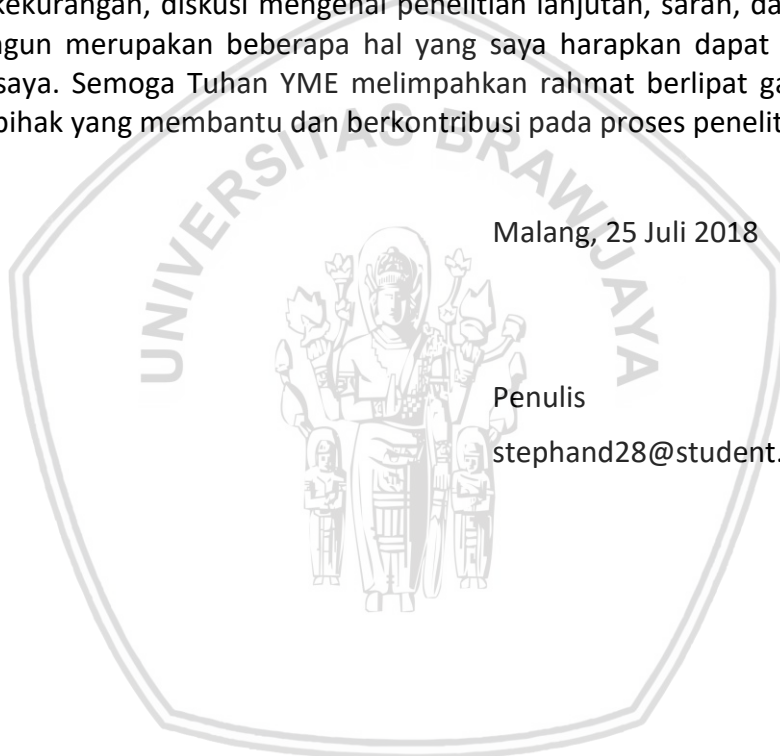
8. Bapak M. Aditya Dharmawan. Amd, Ibu dr. Agustin, Ibu Try Wulandari S.I.Kom, Sdri. Destina Martha Siwi, AMKL dan segenap pegawai seksi rehabilitasi BNN Kota Malang yang memberikan kemudahan pada proses penelitian di lokasi studi kasus.
9. Teman-teman Jurusan Sistem Informasi, Fakultas Ilmu Komputer, yang tidak dapat saya sebutkan namanya satu per satu, yang telah memberikan saya dukungan moril, menjadi teman dan sahabat yang baik dan menyenangkan selama masa perkuliahan.
10. Semua pihak yang tidak bisa penulis sebutkan satu per satu yang telah membantu penulis dalam menyelesaikan skripsi ini.

Tidak lupa saya sampaikan bahwa penelitian yang telah dilakukan memiliki banyak kekurangan, diskusi mengenai penelitian lanjutan, saran, dan kritik yang membangun merupakan beberapa hal yang saya harapkan dapat disampaikan kepada saya. Semoga Tuhan YME melimpahkan rahmat berlipat ganda kepada seluruh pihak yang membantu dan berkontribusi pada proses penelitian ini.

Malang, 25 Juli 2018

Penulis

stephand28@student.ub.ac.id



ABSTRAK

Badan Narkotika Nasional (BNN) Kota Malang merupakan Lembaga Pemerintah vertikal yang berperan dalam Pencegahan, Pemberantasan, Penyalahgunaan Peredaran Gelap Narkoba (P4GN) yang melaksanakan tugasnya berada di bawah tanggung jawab Pemerintah Daerah Kota Malang. Seksi Rehabilitasi bertugas untuk memulihkan para pengguna narkoba, dalam melakukan tugasnya Seksi Rehabilitasi menyimpan data secara konvensional menggunakan kertas formulir. Masalahnya adalah data pasien yang semakin banyak lalu masih disimpan dengan cara konvensional membuat pencarian data menjadi lama berdampak mengurangi produktivitas pegawai sehingga pemeliharaan data menjadi lebih rumit dan risiko kehilangan data fisik menjadi lebih besar, penggunaan sistem informasi yang dikembangkan terbukti dapat mengurangi waktu pencarian data sehingga meningkatkan produktivitas dan efektivitas. Keberhasilan pengembangan sistem informasi tidak lepas dari penggunaan metode pengembangan yang tepat, kebutuhan pemangku kepentingan yang belum terlihat jelas dan sistem yang harus bisa beradaptasi dengan kebutuhan organisasi sementara dikembangkan menjadi alasan kami menggunakan strategi Iterasi Inkremental.

Penelitian ini akan melaporkan penerapan strategi ini serta peran penting lainnya, antara lain analisis persyaratan yang tepat dan perancangan serta metode pengujian yang sesuai dan dapat diukur dengan permasalahan yang sesungguhnya yang menghasilkan 10 persyaratan fungsional, 2 persyaratan nonfungsional dan 8 *use case* pada tahap analisis persyaratan, 10 kelas analisis, 9 tabel, 4 model dan 6 *controller* pada tahap perancangan lalu 3 kasus uji pengujian skenario yang berhasil dalam tahap pengujian.

Kata kunci: data pasien, iterasi inkremental, pengujian skenario, OOAD, *mvc*, *codeigniter*

ABSTRACT

Narcotics National Agency (BNN Kota Malang) is government vertical agencies whose have role in prevention, eradication of Illicit Drug Abuse is doing their job under the responsibility of the local Government of Malang City. the Rehabilitation Section's duty is recovering drug users, Rehabilitation Section storing data using paper forms. The problem is that a growing number of patient data, which stored with the conventional ways so the data search becomes longer and reduce the productivity of the employees so that the maintenance of the data becomes more complex and increase the risk of losing the physical data, information systems proven can reduce the time thus increasing productivity and effectiveness. The success of information system development depends on proper development method, the stakeholder's needs are unclear yet and the system should be able to adapt to the needs of the organization while developed are the reason we use a strategy of Incremental Iteration.

This research will report the implementation of the strategy as well as other important roles, among others is the proper requirements analysis, design and the appropriate testing methods which can be measured with the real problems which result 10 functional requirements and 2 nonfunctional requirements in requirement analysis, 10 class (analysis), 9 table, 4 model, and 6 controller in design phase, 3 test case succeed in testing phase.

Keywords: patient data, iterative incremental, scenario testing, OOAD, mvc, co-deigniter

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xvi
DAFTAR LAMPIRAN	xix
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Gambaran organisasi	6
2.2.1 Visi dan Misi	7
a. Visi	7
b. Misi	7
2.2.2 Struktur organisasi	8
2.3 <i>Business Process Modelling</i> (BPM)	8
2.3.1 <i>Business Process Modelling Notation 2.0</i>	8
2.3.2 Elemen-elemen BPMN 2.0.	8
2.4 Proses Bisnis.....	13
2.4.1 Analisis dan Pemodelan Proses Bisnis <i>As-Is</i>	13
2.4.2 Analisis dan Pemodelan Proses Bisnis <i>To-Be</i>	14
2.5 Analisis Persyaratan	14
2.5.1 Identifikasi Tipe Pemangku Kepentingan.....	14

2.5.2 Analisis Masalah	14
2.5.3 Identifikasi Kebutuhan Pemangku Kepentingan	15
2.5.4 Identifikasi Fitur	15
2.5.5 Identifikasi Persyaratan Fungsional dan Nonfungsional	15
2.5.6 Pemodelan <i>Use Case</i>	16
2.6 <i>Unified Modelling Language</i> (UML)	16
2.6.1 <i>Use-Case Diagram</i>	17
2.6.2 <i>Activity Diagram</i>	18
2.6.3 <i>Class Diagram</i>	20
2.6.4 <i>Sequence Diagram</i>	21
2.7 <i>Physical Data Model</i> (PDM)	22
2.7.1 Karakteristik umum dari <i>Physical Data Model</i> :	22
2.8 <i>Model View Controller</i> (MVC)	23
2.9 Analisis Inkremental & Iterasi	24
2.10 Pengujian	24
2.10.1 <i>Scenario Testing</i>	24
2.10.2 Pengujian <i>Usability</i> menggunakan <i>USE QUESTIONNAIRE</i>	25
2.10.3 Skala <i>Likert</i>	27
BAB 3 METODOLOGI	28
3.1 Studi Literatur	29
3.2 Analisis Persyaratan	29
3.3 Perancangan Sistem	29
3.4 Analisis Inkremental & Iterasi	30
3.5 Implementasi	30
3.6 Pengujian	30
3.7 Diskusi dengan pemangku kepentingan	31
3.8 Penulisan Laporan	31
BAB 4 ANALISIS PERSYARATAN	32
4.1 Pemodelan Proses Bisnis	32
4.1.1 Proses Bisnis <i>As-Is</i>	32
4.1.2 Proses Bisnis <i>To-Be</i>	34
4.2 Analisis Persyaratan	37

4.2.1 Identifikasi Pemangku Kepentingan	37
4.2.2 Analisis Masalah	38
4.2.3 Identifikasi Kebutuhan Pengguna	38
4.2.4 Identifikasi Pengguna	40
4.2.5 Identifikasi Fitur	41
4.2.6 Persyaratan Fungsional	42
4.2.7 Persyaratan Non-fungsional.....	43
4.3 Pemodelan <i>Use Case</i>	44
4.3.1 <i>Use Case</i> Diagram.....	45
4.3.2 Deskripsi Aktor	46
4.3.3 Spesifikasi <i>Use Case</i>	46
4.4 Pemodelan aktivitas.....	55
4.4.1 Pemodelan aktivitas Masuk Sistem	55
4.4.2 Pemodelan aktivitas Mengelola Data Pasien	56
4.4.3 Pemodelan aktivitas Mengelola Data Konseling Pasien	57
4.4.4 Pemodelan aktivitas Cetak Formulir	58
4.4.5 Pemodelan aktivitas Ganti <i>Password</i>	59
4.4.6 Pemodelan aktivitas Menambah Data Konseling	60
4.4.7 Pemodelan aktivitas Menambah Data Pribadi	61
4.4.8 Pemodelan aktivitas Keluar Sistem	62
BAB 5 PERANCANGAN.....	63
5.1 Pemodelan Interaksi	63
5.1.1 <i>Sequence Diagram</i> Masuk Sistem	63
5.1.2 Mengelola data pasien	64
5.1.3 <i>Sequence Diagram</i> Cetak formulir	65
5.1.4 Mengelola data konseling pasien	66
5.1.5 <i>Sequence Diagram</i> Menambah data konseling	67
5.1.6 <i>Sequence Diagram</i> Menambah data pribadi	68
5.1.7 <i>Sequence Diagram</i> Keluar Sistem.....	68
5.2 Pemodelan Objek.....	69
5.2.1 Diagram Kelas Analisis.....	69
5.2.2 Diagram Kelas Perancangan.....	69

5.3 Perancangan Basis Data	74
5.4 Perancangan Algoritme	75
5.4.1 Masuk Sistem	75
5.4.2 Mengelola data pasien	75
5.4.3 Cetak formulir	75
5.4.4 Mengelola data konseling pasien	76
5.4.5 Menambah data konseling pasien	76
5.4.6 Menambahkan data pribadi.....	77
5.4.7 Ganti <i>password</i>	77
5.4.8 Keluar Sistem.....	78
5.5 Perancangan Antarmuka Pengguna	78
5.5.1 Antarmuka Masuk sistem	78
5.5.2 Antarmuka <i>Dashboard</i>	79
5.5.3 Antarmuka Mengelola data pasien.....	79
5.5.4 Antarmuka Mengelola data konseling pasien	80
5.5.5 Antarmuka Cetak formulir	81
5.5.6 Antarmuka Ganti <i>password</i>	81
5.5.7 Antarmuka Menambahkan data pribadi.....	82
5.6 Analisis Inkremental & Iterasi.....	82
5.7 Perancangan Pengujian	85
5.7.1 Perancangan Pengujian Skenario.....	86
5.7.2 Pengujian <i>Usability</i>	88
5.7.3 Pengujian Kompatibilitas	90
BAB 6 IMPLEMENTASI	91
6.1 Spesifikasi Lingkungan Implementasi	91
6.2 Implementasi Algoritme	92
6.2.1 Masuk Sistem	92
6.2.2 Mengelola data pasien.....	93
6.2.3 Cetak formulir	94
6.2.4 Mengelola data konseling pasien	95
6.2.5 Menambah data konseling pasien	95
6.2.6 Menambahkan data pribadi.....	96

6.2.7 Ganti <i>password</i>	97
6.2.8 Keluar Sistem.....	98
6.3 Implementasi Antarmuka pengguna	98
6.3.1 Antarmuka Masuk sistem	98
6.3.2 Antarmuka <i>Dashboard</i>	99
6.3.3 Antarmuka Mengelola data pasien.....	99
6.3.4 Antarmuka Mengelola data konseling pasien	100
6.3.5 Antarmuka Cetak formulir	100
6.3.6 Antarmuka Ganti <i>password</i>	101
6.3.7 Antarmuka Menambahkan data pribadi.....	101
BAB 7 PENGUJIAN	102
7.1 Pengujian Skenario	102
7.1.1 Hasil Pengujian Skenario	102
7.1.2 Kesimpulan Pengujian Skenario	105
7.2 Pengujian <i>Usability</i>	105
7.2.1 Hasil Pengujian <i>Usability</i>	106
7.2.2 Kesimpulan Pengujian <i>Usability</i>	108
7.3 Pengujian Kompatibilitas	109
7.3.1 Hasil Pengujian Kompatibilitas.....	109
7.3.2 Kesimpulan Pengujian Kompatibilitas <i>Browser</i>	109
BAB 8 Penutup	110
8.1 Kesimpulan.....	110
8.2 Saran	111
Daftar Pustaka.....	112

DAFTAR TABEL

Tabel 2.1 Kajian Pustaka 1.....	5
Tabel 2.2 Kajian Pustaka 2.....	5
Tabel 2.3 Kajian Pustaka 3.....	6
Tabel 2.4 Kerangka Dokumentasi Pernyataan Masalah	15
Tabel 2.5 Kerangka Tabel fitur	15
Tabel 2.6 Format Spesifikasi <i>Use Case</i>	16
Tabel 2.7 Simbol <i>Use Case Diagram</i>	17
Tabel 2.8 Simbol <i>Activity Diagram</i>	18
Tabel 2.9 Simbol <i>Class Diagram</i>	20
Tabel 2.10 Simbol <i>Sequence Diagram</i>	21
Tabel 2.11 Penilaian jawaban kuantitatif.....	27
Tabel 4.1 Perbandingan waktu proses bisnis <i>to-be</i> dan <i>as-is</i>	35
Tabel 4.2 Perubahan aktivitas Proses Bisnis Penerimaan Klien Baru	35
Tabel 4.3 Perbandingan waktu proses bisnis <i>to-be</i> dan <i>as-is</i>	36
Tabel 4.4 Perubahan aktivitas Proses Bisnis Cari Data	37
Tabel 4.5 Tipe Pemangku Kepentingan.....	37
Tabel 4.6 <i>Problem Statement</i>	38
Tabel 4.7 Hasil Identifikasi Kebutuhan Pengguna.....	39
Tabel 4.8 Hasil Identifikasi Pengguna	40
Tabel 4.9 Hasil Identifikasi Fitur	41
Tabel 4.10 Hubungan Kebutuhan Pengguna dan Fitur	42
Tabel 4.11 Persyaratan Fungsional	42
Tabel 4.12 Persyaratan Non-fungsional.....	43
Tabel 4.13 Hubungan aktivitas Proses Bisnis <i>to-be</i> dengan <i>Use Case</i>	45
Tabel 4.14 Hubungan <i>Use Case</i> dan Pemangku Kepentingan	45
Tabel 4.15 Hubungan <i>Use Case</i> dan Fitur	46
Tabel 4.16 Deskripsi Aktor	46
Tabel 4.17 Spesifikasi <i>Use Case</i> Masuk Sistem	47
Tabel 4.18 Spesifikasi <i>Use Case</i> Mengelola Data Pasien.....	47
Tabel 4.19 Spesifikasi <i>Use Case</i> Mengelola Data Konseling Pasien	48
Tabel 4.20 Spesifikasi <i>Use Case</i> Cetak Formulir	50

Tabel 4.21 Spesifikasi <i>Use Case</i> Ganti Password.....	51
Tabel 4.22 Spesifikasi <i>Use Case</i> Menambah Data Konseling	52
Tabel 4.23 Spesifikasi <i>Use Case</i> Menambah Data Pribadi	53
Tabel 4.24 Spesifikasi <i>Use Case</i> Keluar Sistem.....	54
Tabel 5.1 <i>Pseudocode</i> Masuk Sistem	75
Tabel 5.2 <i>Pseudocode</i> Mengelola Data Pasien	75
Tabel 5.3 <i>Pseudocode</i> Cetak Formulir.....	76
Tabel 5.4 <i>Pseudocode</i> Mengelola data konseling pasien	76
Tabel 5.5 <i>Pseudocode</i> Menambah data konseling pasien	77
Tabel 5.6 <i>Pseudocode</i> Menambahkan data pribadi.....	77
Tabel 5.7 <i>Pseudocode</i> Ganti Password	77
Tabel 5.8 <i>Pseudocode</i> Keluar Sistem.....	78
Tabel 5.9 Analisis Inkremental & Iterasi Pengembangan SmartRehab	82
Tabel 5.10 Kasus Uji Skenario Pendaftaran Pasien Baru	86
Tabel 5.11 Kasus Uji Skenario Pencarian berdasarkan kategori	87
Tabel 5.12 Kasus Uji Skenario Menambah data konseling	87
Tabel 5.13 Rancangan Kuesioner untuk menguji <i>Usability</i>	88
Tabel 5.14 Perancangan Kasus Uji Kompatibilitas	90
Tabel 6.1 Spesifikasi Perangkat Keras	91
Tabel 6.2 Spesifikasi Perangkat Lunak	91
Tabel 6.3 Spesifikasi Lingkungan Instalasi.....	92
Tabel 6.4 Kode Program Masuk Sistem	93
Tabel 6.5 Kode Program Mengelola Data Pasien.....	93
Tabel 6.6 Kode Program Cetak Formulir	94
Tabel 6.7 Kode Program Mengelola data konseling pasien.....	95
Tabel 6.8 Kode Program Menambah data konseling pasien	95
Tabel 6.9 Kode Program Menambahkan data pribadi.....	96
Tabel 6.10 Kode Program Ganti password.....	97
Tabel 6.11 Kode Program Keluar Sistem.....	98
Tabel 7.1 Hasil Pengujian Skenario Menambah data pribadi	103
Tabel 7.2 Hasil Pengujian Skenario Pencarian berdasarkan kategori.....	104
Tabel 7.3 Hasil Pengujian Skenario menambah data konseling	105

Tabel 7.4 Interpretasi skor <i>Likert</i>	106
Tabel 7.5 Hasil Pengujian <i>Usability</i>	106
Tabel 7.6 Analisis Pengujian <i>Usability</i>	108



DAFTAR GAMBAR

Gambar 2.1 Struktur organisasi BNN Kota Malang.....	8
Gambar 2.2 Jenis-jenis <i>Event</i>	9
Gambar 2.3 <i>Activities</i>	9
Gambar 2.4 <i>Collapsed Sub-Process</i>	10
Gambar 2.5 <i>Expanded Sub-Process</i>	10
Gambar 2.6 <i>Gateways</i>	10
Gambar 2.7 Data	11
Gambar 2.8 <i>Connecting Objects</i>	11
Gambar 2.9 <i>Message Flows</i>	11
Gambar 2.10 <i>Association</i>	11
Gambar 2.11 Data Association.....	12
Gambar 2.12 <i>Swimlanes</i>	12
Gambar 2.13 <i>Artifacts</i>	13
Gambar 2.14 Contoh <i>use case diagram</i>	18
Gambar 2.15 Contoh <i>Activity Diagram</i>	19
Gambar 2.16 Contoh <i>Class Diagram</i>	21
Gambar 2.17 Contoh <i>Sequence Diagram</i>	22
Gambar 2.18 Contoh <i>Physical Data Model</i>	23
Gambar 2.19 Konsep <i>Model-view-controller</i>	23
Gambar 3.1 Diagram Metodologi Penelitian	28
Gambar 4.1 Proses Bisnis <i>As-Is</i> Penerimaan Pasien Baru	33
Gambar 4.2 Proses Bisnis <i>As-Is</i> Mencari Data Pasien	34
Gambar 4.3 Proses Bisnis <i>To-Be</i> Penerimaan Pasien Baru	35
Gambar 4.4 Proses Bisnis <i>To-Be</i> Mencari data pasien	36
Gambar 4.5 Keterangan Kodifikasi Kebutuhan Pengguna	38
Gambar 4.6 Kodifikasi fitur	41
Gambar 4.7 Kodifikasi Persyaratan Fungsional.....	42
Gambar 4.8 Kodifikasi Persyaratan Non-fungsional	43
Gambar 4.9 <i>Use-Case Diagram</i> Sistem Informasi “SmartRehab”	44
Gambar 4.10 <i>Activity Diagram</i> Masuk Sistem	55
Gambar 4.11 <i>Activity Diagram</i> Mengelola Data Pasien.....	56

Gambar 4.12 <i>Activity Diagram</i> Mengelola Data Konseling Pasien	57
Gambar 4.13 <i>Activity Diagram</i> Cetak Formulir	58
Gambar 4.14 <i>Activity Diagram</i> Ganti <i>Password</i>	59
Gambar 4.15 <i>Activity Diagram</i> Menambah Data Konseling	60
Gambar 4.16 <i>Activity Diagram</i> Menambah Data Pribadi.....	61
Gambar 4.17 <i>Activity Diagram</i> Keluar Sistem	62
Gambar 5.1 <i>Sequence Diagram</i> Masuk Sistem	63
Gambar 5.2 <i>Sequence Diagram</i> Menampilkan daftar pasien	64
Gambar 5.3 <i>Sequence Diagram</i> Menampilkan detail pasien	64
Gambar 5.4 <i>Sequence Diagram</i> Mengubah data pasien.....	65
Gambar 5.5 <i>Sequence Diagram</i> Cetak Formulir	65
Gambar 5.6 <i>Sequence Diagram</i> Menampilkan daftar pasien konseling	66
Gambar 5.7 <i>Sequence Diagram</i> Menampilkan detail konseling	66
Gambar 5.8 <i>Sequence Diagram</i> Mengubah data konseling	67
Gambar 5.9 <i>Sequence Diagram</i> Menambah data konseling.....	67
Gambar 5.10 <i>Sequence Diagram</i> Menambah data pribadi.....	68
Gambar 5.11 <i>Sequence Diagram</i> Keluar Sistem	68
Gambar 5.12 Diagram Kelas Analisis.....	71
Gambar 5.13 Diagram Kelas Perancangan <i>Model</i>	72
Gambar 5.14 Diagram Kelas Perancangan <i>Controller</i>	73
Gambar 5.15 Diagram Kelas Perancangan Relasi <i>Controller & Model</i>	73
Gambar 5.16 <i>Physical Data Model</i> SmartRehab.....	74
Gambar 5.17 Sketsa Tampilan Antarmuka Masuk Sistem	79
Gambar 5.18 Sketsa Tampilan Antarmuka Dashboard.....	79
Gambar 5.19 Sketsa Tampilan Antarmuka Mengelola data pasien.....	80
Gambar 5.20 Sketsa Tampilan Antarmuka Mengelola data konseling pasien	80
Gambar 5.21 Sketsa Tampilan Antarmuka Cetak Formulir	81
Gambar 5.22 Sketsa Tampilan Antarmuka Ganti <i>Password</i>	81
Gambar 5.23 Sketsa Tampilan Antarmuka Menambahkan data pribadi	82
Gambar 5.24 Kodefikasi Kasus Uji.....	86
Gambar 6.1 Antarmuka Masuk Sistem	98
Gambar 6.2 Antarmuka <i>Dashboard</i>	99

Gambar 6.3 Antarmuka Mengelola data pasien	99
Gambar 6.4 Antarmuka Mengelola data konseling pasien.....	100
Gambar 6.5 Antarmuka Cetak Formulir	100
Gambar 6.6 Antarmuka Ganti <i>Password</i>	101
Gambar 6.7 Antarmuka Menambahkan data pribadi.....	101
Gambar 7.1 Hasil Pengujian Kompatibilitas.....	109
Gambar 7.2 Skor Kompatibilitas	109



DAFTAR LAMPIRAN

LAMPIRAN A RINGKASAN WAWANCARA VALIDASI MASALAH	115
LAMPIRAN B CONTOH USE KUESIONER.....	117



BAB 1 PENDAHULUAN

1.1 Latar belakang

Badan Narkotika Nasional Kota Malang (BNN) merupakan badan negara vertikal yang melaksanakan tugas, fungsi, wewenang dari kepala BNN dan bertanggung jawab pada BNN provinsi dalam melakukan P4GN (Program Pemberantasan Penyalahgunaan dan Peredaran Gelap Narkoba). BNN Kota Malang terdiri dari atas kepala, subbagian umum, seksi pencegahan dan pemberdayaan masyarakat, seksi rehabilitasi dan seksi pemberantasan ("Peraturan Kepala BNN No. 3 Tahun 2015," 2015).

Berdasarkan hasil wawancara terhadap seksi rehabilitasi BNN Kota Malang, mereka mengalami kesulitan dalam mencari data yang diarsipkan (*hardcopy*), sebagai contoh jika BNN-P (Tingkat Provinsi) ingin mengetahui data berapa banyak remaja putri yang sedang menjalani rehabilitasi di BNN Kota Malang maka seksi rehabilitasi akan menelusuri satu persatu data yang diarsipkan yang berupa *hardcopy* lalu mengidentifikasi dan menghitung secara manual, lalu penyimpanan data pasien yang berupa *hardcopy* sangat berisiko hilang atau rusak jika terjadi musibah atau bencana yang tidak bisa diperkirakan, sehingga pembuatan sistem informasi untuk data pasien adalah solusi untuk mempercepat pencarian data dan memelihara data (Jervis & Masoodian, 2014)

Sistem yang akan dikembangkan diharapkan dapat menjalankan beberapa kebutuhan fungsional sebelum semua kebutuhan selesai dikembangkan, karena itu penulis menggunakan metode *Iterative Incremental*, yang dapat menghasilkan sistem yang berevolusi sedikit demi sedikit (*increment*) yang di dalamnya menjalankan iterasi, sampai semua kebutuhan dipenuhi (Cockburn, 2008). Sistem juga diharapkan dapat mudah digunakan, sehingga demo di depan para pengguna sekaligus dapat membuat para pengguna nantinya terbiasa, dan dalam pengembangan *iterative incremental*, pengujian bukan akhir dari siklus pengembangan tapi dilakukan hingga sistem dapat sesuai dengan kebutuhan pemangku kepentingan (Ramadan & Sodermark, 2014)

Di dalam tahap analisis perancangan penulis menggunakan pendekatan OOAD (*Object Oriented Analysis and Design*) dengan menggunakan permodelan UML untuk mempermudah pemahaman dan komunikasi (Fowler, 2004). Data pasien yang memiliki banyak berkas dan data yang saling berhubungan menjadi alasan menggunakan OOAD (Rickman & Company, 2001). Mengembangkan sistem dengan berorientasi objek diharapkan dapat mempermudah kelanjutan sistem ini, dengan pengalaman dan hasil-hasil yang sudah terbukti hingga saat ini *object oriented* sangat sering digunakan dan dipadukan dengan UML yang juga terbukti berhasil berguna dalam menunjang pengembangan berorientasi objek ini (M. Fowler, 1997)

Tahap pengujian menggunakan *Scenario Testing*. Menurut (Kaner, 2003a), *Scenario testing* berguna untuk mencari kesalahan (*bugs*) pada sistem sebanyak-banyaknya dan juga berguna untuk mengeksplorasi penggunaan ahli dari sistem,

yang bisa berarti bagaimana menggunakan sistem secara maksimal seperti yang dibutuhkan oleh *stakeholder*. Pengujian Skenario mendukung metode pengembangan *Iterative Incremental* dimana ketika dalam tahap pengujian lalu menemukan kesalahan maka kesalahan tersebut dapat diatasi dengan menjalankan iterasi untuk memperbaiki kesalahan tersebut.

Berdasarkan deskripsi di atas maka diperlukan pengembangan sistem. Sistem yang akan dikembangkan adalah pelaporan hasil kegiatan seksi rehabilitasi, karena sistem ini sangat dibutuhkan. Oleh karena itu penulis mengangkat penelitian yang berjudul ***“Pengembangan Sistem Informasi Data Pasien Seksi Rehabilitasi BNN Kota Malang Menggunakan Metode Iterative Incremental”***

1.2 Rumusan masalah

Berdasarkan latar belakang di atas maka dapat dirumuskan rumusan masalah sebagai berikut:

1. Bagaimana hasil analisis persyaratan dari sistem informasi Data Pasien Seksi Rehabilitasi BNN Kota Malang?
2. Bagaimana penerapan OOAD (*Object Oriented Analysis Design*) dalam merancang sistem informasi Data Pasien Seksi Rehabilitasi BNN Kota Malang?
3. Bagaimana penerapan metode *Iterative Incremental* dalam mengembangkan sistem informasi Data Pasien Seksi Rehabilitasi BNN Kota Malang?
4. Bagaimana hasil pengujian dari sistem informasi Data Pasien Seksi Rehabilitasi BNN Kota Malang?

1.3 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah:

Tujuan umum:

Mengembangkan sistem informasi Data Pasien Seksi Rehabilitasi BNN Kota Malang

Tujuan khusus:

1. Mengidentifikasi persyaratan fungsional dan non fungsional sistem
2. Merancang sistem dengan permodelan berorientasi objek
3. Mengimplementasikan sistem dengan teknologi berorientasi obyek
4. Menguji sistem sesuai dengan persyaratan fungsional dan non fungsionalnya

1.4 Manfaat

1. Bagi BNN Kota Malang
 - a. Mendapatkan sistem informasi pelaporan kegiatan bagi seksi rehabilitasi yang diharapkan dapat menunjang operasional seksi rehabilitasi

2. Bagi Penulis

- a. Suatu kebahagiaan tersendiri bagi penulis karena bisa memberikan solusi permasalahan menggunakan keterampilan secara ilmiah dan dapat diukur
- b. Menambah wawasan dan keterampilan untuk mengembangkan sistem, yaitu keterampilan untuk menganalisis, merancang, menerapkan hingga menguji suatu sistem informasi

3. Bagi Pembaca

- a. Menjadikan penelitian ini sebagai referensi penelitian selanjutnya
- b. Menambah wawasan dalam pengembangan sistem
- c. Menambah keingintahuan tentang metode dalam pengembangan sistem

1.5 Batasan masalah

Berdasarkan rumusan masalah di atas maka batasan masalah sebagai berikut:

1. Pengembangan sistem dilakukan dalam lingkup BNN Kota Malang dan terbatas hanya seksi rehabilitasi saja
2. Metode yang digunakan adalah *Iterative Incremental*
3. *Framework* yang digunakan adalah *CodeIgniter* (php) dengan DBMS mysql
4. Penelitian ini hanya sampai pengembangan sistem, pengujian yang dilakukan adalah *Scenario Testing*

1.6 Sistematika pembahasan

BAB 1

PENDAHULUAN

Pada bab ini berisi latar belakang, rumusan masalah, tujuan dan manfaat, serta batasan masalah tentang pengembangan sistem pelaporan kegiatan

BAB 2

LANDASAN KEPUSTAKAAN

Pada bab ini berisi ulasan penelitian-penelitian yang menginspirasi penulis yang mempunyai metode yang sama, dan juga berisi teori/dasar yang digunakan serta literatur ilmiah lainnya yang berhubungan dan mendukung dalam pengembangan sistem informasi data pasien

BAB 3 METODE PENELITIAN

Pada bab ini dijelaskan alur penelitian dalam menyelesaikan pengembangan sistem yang berisi tahap dalam SDLC.

BAB 4 ANALISIS PERSYARATAN

Pada bab ini dijelaskan bagaimana mengumpulkan data sampai mengolah data menjadi hasil analisis (daftar kebutuhan sistem) dan memodelkannya (*Use Case, Activity Diagram*) agar dapat dipahami oleh kedua pihak (pemangku kepentingan dan peneliti).

BAB 5 PERANCANGAN SISTEM

Pada bab ini terdapat diagram-diagram rancangan (*sequence diagram & class diagram*), *Physical data model* sistem dan prototipe sistem, juga terdapat hasil analisis *incremental* untuk menentukan urutan fitur-fitur atau bagian yang diimplementasi yang merupakan acuan untuk mengimplementasikan sistem.

BAB 6 IMPLEMENTASI

Pada bab ini dimulai dengan spesifikasi lingkungan pengembangan lalu implementasi kode sistem dari hasil analisis dan perancangan sistem dan *user interface* sistem.

BAB 7 PENGUJIAN

Pada bab ini menguji apakah sistem yang dikembangkan sudah menjawab daftar kebutuhan sistem dan *Scenario Testing* untuk menguji skenario tertentu dalam menggunakan sistem ini.

BAB 8 PENUTUP

Pada bab ini berisi kesimpulan hasil penelitian yang merupakan ringkasan yang berisi hasil dan deskripsi untuk menjawab rumusan masalah dalam mengembangkan sistem dan saran yang muncul berdasarkan kekurangan dari penelitian ini untuk penelitian selanjutnya

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Tabel 2.1 Kajian Pustaka 1

Masalah	Metode	Hasil
Bagaimana cara terbaik mengimplementasikan Sistem Rekam Medis yang besar, efisien, dan <i>user friendly</i> yang akan meningkatkan keseragaman informasi pasien (secara efisien dapat menyimpan dan melihat data dengan aman)	Penelitian dari (Ramadan & Sodermark, 2014) yang berjudul "Development of an Electronic Medical Record System" menggunakan metode pengembangan Iterative Incremental	Model kerja iteratif mengarah pada pengembangan awal produk (tes dan jalankan). Dengan sedikit perubahan kecil, peneliti dapat mengidentifikasi <i>bug</i> dengan mudah pada versi baru. Model ini telah bekerja dengan baik untuk jenis proyek ini yang membutuhkan fleksibilitas dan adaptasi yang memiliki persyaratan tambahan datang selama fase implementasi

Tabel 2.2 Kajian Pustaka 2

Masalah	Metode	Hasil
Amerika Serikat membutuhkan cara yang lebih efektif untuk mendeteksi dan memantau kebakaran hutan yang berpotensi berbahaya. (Implementasi Misi FireSAT)	Penelitian dari (Cloutier & Griego, 2008) yang berjudul " <i>Applying Object Oriented Systems Engineering to Complex Systems</i> " Menggunakan pemodelan OOAD (<i>Use Case</i> dan <i>Activity Diagram</i>)	Penulis menggunakan <i>Use Case</i> dan <i>Activity Diagram</i> kemudian dianalisis untuk menghasilkan <i>Logical Architecture</i> . Menghasilkan sistem yang modular, sistemnya memang kompleks, tetapi dengan pendekatan OOAD, data dan proses dapat menjadi 1 blok (class)

Tabel 2.3 Kajian Pustaka 3

Masalah	Metode	Hasil
Para <i>stakeholder</i> (seperti staf penjualan atau staf <i>helpdesk</i>) mungkin mendukung produk dengan tidak antusias; pelanggan mungkin kurang bersedia membayar untuk itu, pengguna akhir mungkin kurang bersedia untuk mengadopsinya. (peringatan dini untuk masalah-masalah <i>requirements</i> yang akan menghantui proyek nantinya) karena <i>requirements</i> tidak menjawab masalah sebenarnya	Penelitian dari (Kaner, 2003a) memperkenalkan metode yaitu " <i>Scenario Testing</i> " dalam penelitiannya yang berjudul " <i>An Introduction to Scenario Testing</i> "	<i>Scenario testing</i> berfungsi paling baik untuk transaksi atau <i>event</i> yang kompleks, untuk mempelajari pengiriman manfaat program secara menyeluruh, untuk menjelajahi bagaimana program akan bekerja di tangan pengguna yang berpengalaman, dan untuk mengembangkan variasi <i>bug</i> yang lebih persuasif yang ditemukan daripada menggunakan pendekatan lain

Berdasarkan Tabel 2.1 dengan penerapan metode *iterative-incremental* proses pengembangan dapat lebih cepat diketahui *bugnya* dan lebih cepat *release*, sesuai dengan apa yang diharapkan oleh *stakeholder*. *Scenario testing* juga membuat pengguna dapat mempelajari sistem, menemukan *bug* yang terdapat jika sistem dijalankan secara *scenario*, tidak hanya menguji per fungsionalnya, karena nantinya sistem ini harus dapat berjalan sesuai skenario (Tabel 2.3) dan penggunaan OOAD dapat memaksimalkan fungsionalitas sistem yang kaitannya sangat erat dengan *data relationship* dan proses yang menjadi satu menjadi sebuah *object*, karena setiap pasien memiliki banyak data dan beberapa proses. (Tabel 2.2)

2.2 Gambaran organisasi

Badan Narkotika Nasional Kota Malang adalah Lembaga Pemerintah Vertikal yang berkedudukan di bawah kendali dan tanggung jawab kepada Badan Narkotika Nasional Provinsi dan Badan Narkotika Nasional. Badan Narkotika Nasional merupakan lembaga yang ditunjuk langsung untuk melakukan Program Pencegahan dan Pemberantasan Penyalahgunaan dan Peredaran Gelap Narkoba (P4GN) dengan berbagai kegiatan melalui Bidang Pencegahan, Bidang Pemberdayaan Masyarakat, Bidang Rehabilitasi dan Bidang Pemberantasan.

Badan Narkotika Kota Malang adalah organisasi yang beranggotakan Pemerintah Daerah Kota Malang dan Polresta Malang yang dipimpin oleh seorang Kepala Badan Narkotika yang dalam melaksanakan tugasnya berada di bawah dan bertanggung jawab kepada Pemerintah Daerah Kota Malang. Badan Narkotika Kota Malang memiliki fungsi antara lain:

1. Pengoordinasian instansi pemerintah di Kota Malang dalam penyiapan dan penyusunan kebijakan di bidang terkait pencegahan dan pemberantasan penyalahgunaan narkotika, psikotropika, prekursor dan zat adiktif lainnya;
2. Pengoordinasian instansi pemerintah terkait di Kota Malang dalam pelaksanaan kebijakan di bidang ketersediaan pencegahan dan pemberantasan penyalahgunaan narkotika, psikotropika, prekursor dan zat adiktif lainnya serta pencegahan permasalahan dalam pelaksanaan tugas;
3. Pengoordinasian instansi pemerintah terkait di Kota Malang dalam kegiatan pengadaan, pengendalian dan pengawasan di bidang narkotika, psikotropika, prekursor dan zat adiktif lainnya;
4. Pemutusan jaringan peredaran gelap narkotika, psikotropika, prekursor dan zat adiktif lainnya;
5. Pelaksanaan kerja sama nasional antar daerah dan di wilayah Kota Malang dalam rangka penanggulangan masalah narkotika, psikotropika, prekursor dan zat adiktif lainnya;
6. Pembangunan dan pengembangan sistem informasi narkotika, psikotropika, prekursor dan zat adiktif lainnya bekerja sama dengan Badan Narkotika Nasional.

2.2.1 Visi dan Misi

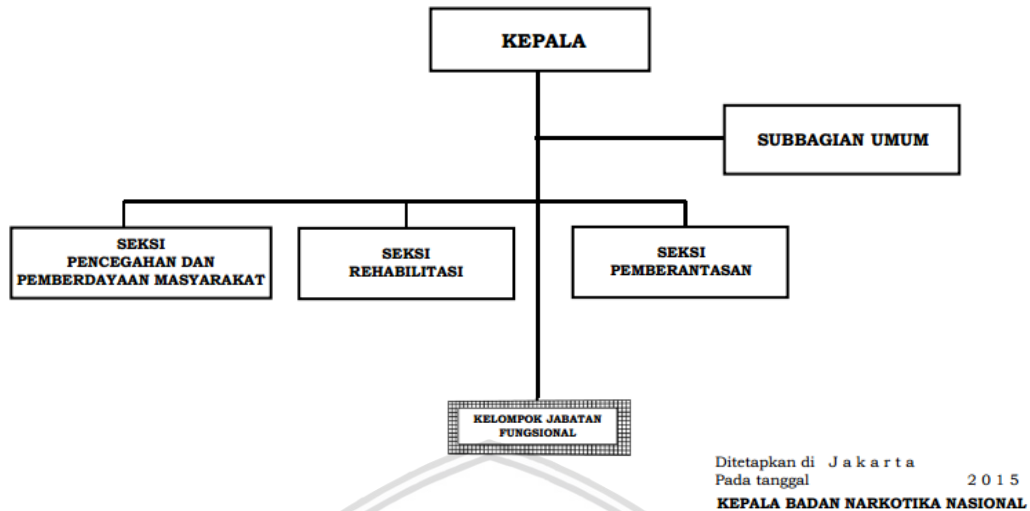
a. Visi

Menjadi perwakilan BNN yang profesional menyatukan dan menggerakkan seluruh komponen masyarakat, instansi pemerintah dan swasta di Kota Malang.

b. Misi

Bersama instansi pemerintah daerah, swasta, dan komponen masyarakat di Kota Malang dengan melaksanakan pencegahan, pemberdayaan masyarakat, penjangkauan dan pendampingan, pemberantasan serta di dukung tata kelola pemerintahan yang akuntabel dalam rangka P4GN.

2.2.2 Struktur organisasi



Gambar 2.1 Struktur organisasi BNN Kota Malang

Sumber: ("Peraturan Kepala BNN No. 3 Tahun 2015," 2015)

2.3 Business Process Modelling (BPM)

Pemodelan proses bisnis adalah sebuah aktivitas untuk memodelkan regulasi atau kegiatan bisnis sebuah organisasi, pemodelan biasanya dilakukan menggunakan diagram, pemodelan proses bisnis berguna untuk memahami dan mengerti proses dari kegiatan bisnis suatu organisasi, pemodelan menggunakan data asli sesuai dengan proses kegiatan bisnis, setelah mengetahui pemodelan proses bisnis diharapkan pembaca dapat mendefinisikan langkah-langkah atau aktivitas usulan yang tepat sesuai tujuan dari organisasi itu sendiri. (Weske, 2012)

2.3.1 Business Process Modelling Notation 2.0

Sebuah *tools* untuk merancang sebuah proses bisnis itulah BPMN. Berisi panduan dan notasi yang digunakan untuk menggambarkan sebuah proses bisnis, penggunaannya sangat luas dan biasa dipakai oleh praktisi dan akademisi, berguna untuk memberikan pemahaman yang satu dan tidak ambigu dengan menggunakan beberapa sudut pandang antar *developer* (*System Analyst, Programmer, Business Analyst*) dan mungkin kepada *stakeholder*, tidak hanya berguna untuk keperluan otomasi penggunaannya bisa bervariasi sesuai kasus dan kebutuhan *stakeholder*. (Weske, 2012)

2.3.2 Elemen-elemen BPMN 2.0.

Lima kategori dasar elemen BPMN adalah (Object Management Group (OMG), 2011):

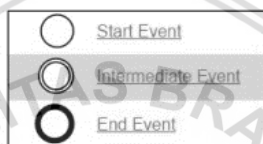
2.3.2.1 Flow Objects

Flow Objects adalah objek utama yang mengalir didalam *pool* atau bisa juga dibatasi dengan *swimlane* yang mendefinisikan perilaku Proses Bisnis. Ada tiga macam *Flow Objects*, yaitu:

Events

Merupakan penyebab/pemicu mengapa sebuah proses terjadi. Ditunjukkan pada Gambar 2.2 (simbol dari *Events*).

- 1) *Start*: Pemicu proses akan dimulai.
- 2) *Intermediate*: Terjadi di dalam *start event* hingga *end event* (opsional), memengaruhi aliran proses tetapi tidak memulai atau langsung mengakhiri.
- 3) *End*: Pemicu proses akan diakhiri



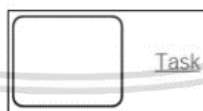
Gambar 2.2 Jenis-jenis Event

Sumber : (Grosskopf, Decker, & Weske, 2009)

Activities

Aktivitas adalah pemodelan sebuah pekerjaan (*task*) yang dilakukan oleh organisasi. Sebuah aktivitas dapat berdiri *atomic* (sendiri) atau berisi gabungan (*non atomic*). Tipe dari *activities* adalah:

- 1) *Task*: Adalah *atomic Activity* (Aktivitas yang berdiri sendiri) yang termasuk dalam sebuah proses.



Gambar 2.3 Activities

Sumber : (Grosskopf et al., 2009)

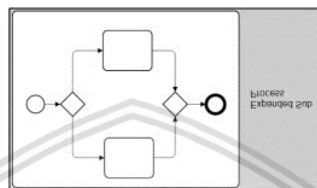
- 2) *Process/Sub-Process*: Merupakan aktivitas yang didalamnya terdiri atas beberapa aktivitas yang dibagi menjadi dua yaitu:
 - a. *Collapsed Sub-Process* adalah detail dari sub-proses yang tidak tampak dalam suatu diagram. Terdapat tanda "*plus*" (+) pada bagian bawah menunjukkan bahwa kegiatan tersebut merupakan sub-proses yang memiliki tingkat yang paling rendah. Notasi *Collapsed Sub-Process* dapat dilihat pada gambar



Gambar 2.4 Collapsed Sub-Process

Sumber : (Grosskopf et al., 2009)

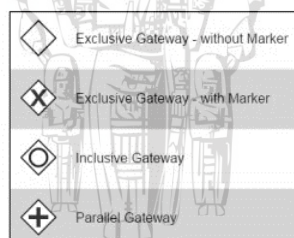
- b. *Expanded Sub-Process* merupakan pemodelan rinci dari sub-proses yang ditunjukkan (*expanded*). Perhatikan, *sequence flow* tidak dapat menyeberangi batas dirinya sendiri. Pada Gambar berikut merupakan contoh dari *Expanded Sub-Process*.



Gambar 2.5 Expanded Sub-Process

Sumber : (Grosskopf et al., 2009)

- c. Gateways digunakan untuk melakukan seleksi kondisi ataupun percabangan dan penggabungan alur yang digambarkan dengan bentuk seperti wajik. Tanda di dalam *Gateway* akan menentukan perilaku seleksi kondisi.



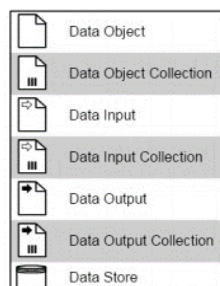
Gambar 2.6 Gateways

Sumber : (Grosskopf et al., 2009)

2.3.2.2 Data

Data digambarkan oleh empat elemen yang pemodelannya dapat ditunjukkan pada Gambar 2.7, yaitu:

- Data Objects* : Adalah mekanisme untuk menunjukkan bagaimana data merupakan hasil atau dibutuhkan aktivitas. *Data object* direlasikan dengan aktivitas yang terkait melalui *Associations*.
- Data Inputs*. Merupakan masukan eksternal untuk proses (parameter).
- Data Outputs*. Merupakan hasil keluaran dari proses (parameter kembalian).
- Data Stores*. Merupakan tempat proses dapat membaca atau menulis data.



Gambar 2.7 Data

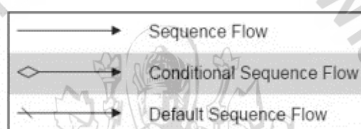
Sumber : (Grosskopf et al., 2009)

2.3.2.3 Connecting Objects

Merupakan notasi untuk memodelkan objek dengan objek lainnya pada proses bisnis:

a. Sequence Flows

Adalah urutan kegiatan yang dilakukan dalam sebuah proses. Macam-macam *sequence flow* ditunjukkan pada gambar

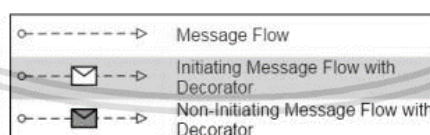


Gambar 2.8 Connecting Objects

Sumber : (Grosskopf et al., 2009)

b. Message Flows

Adalah tanda untuk sebuah *flow* pesan antara dua entitas/objek yang siap untuk mengirim dan menerima. Tipe *message flows* dapat dilihat pada Gambar 2.9.

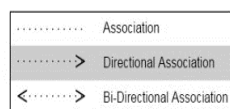


Gambar 2.9 Message Flows

Sumber : (Grosskopf et al., 2009)

c. Associations

Berguna untuk menghubungkan informasi (elemen) terkait artefak. Notasi *Association* ditunjukkan pada gambar.



Gambar 2.10 Association

Sumber : (Grosskopf et al., 2009)

d. *Data Associations*

Sebuah mata dari panah *Association* yang berguna untuk menunjukkan arah aliran.



Gambar 2.11 Data Association

Sumber : (Grosskopf et al., 2009)

2.3.2.4 Swimlanes

Swimlanes dinotasikan dengan sebuah garis yang memisahkan dan mengelompokkan aktor. *Swimlanes* bertanggung jawab membagi peran aktor yang memodelkan kemampuan fungsional atau tanggung jawab yang berbeda. Ada dua cara untuk mengelompokkan elemen model utama melalui "*Swimlanes*" yang contoh gambar nya dapat dilihat pada gambar, yaitu:

- Pools*. Merepresentasikan aktor atau sistem apa saja dalam sebuah proses.
- Lanes*. Merupakan sub-bagian dalam sebuah *pool*. *Lanes* digunakan untuk mengatur dan memberikan klasifikasi aktivitas.



Gambar 2.12 Swimlanes

Sumber : (Grosskopf et al., 2009)

2.3.2.5 Artifacts

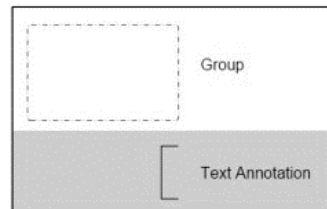
Artifacts berperan dalam memberikan informasi tambahan tentang sebuah proses. BPMN dirancang agar para perancang dan *business analyst* menggunakan alat pemodelan yang lebih fleksibel untuk memperluas notasi dasar serta menyediakan kemampuan untuk beberapa konteks tambahan yang tepat pada kasus atau kondisi tertentu. Berbagai *Artifacts* dapat ditambahkan ke dalam diagram sesuai dengan konteks dan kasus dari proses bisnis yang dimodelkan itu sendiri.

Pemodelan notasi *Artifacts* dapat dilihat pada gambar. Tipe notasi *Artifacts* yang ada meliputi:

- Group*. Merupakan penggabungan *elements* yang memiliki kategori dengan klasifikasi yang sama. Mampu digunakan untuk dokumentasi atau analisis,

tetapi tidak memengaruhi *sequence flow*. Dinotasikan dengan bentuk persegi panjang yang digambarkan dengan *dash* (garis putus-putus).

- b. *Text Annotation*. Adalah Mekanisme bagi perancang memberikan informasi teks tambahan atau keterangan detail untuk pembaca diagram BPMN.



Gambar 2.13 Artifacts

Sumber : (Grosskopf et al., 2009)

2.4 Proses Bisnis

Proses bisnis merupakan gabungan alur aktivitas yang membutuhkan sebuah sampai lebih masukkan yang lalu di proses dan menghasilkan hasil, aktivitas-aktivitas tersebut menghasilkan nilai lebih bagi pemangku kepentingan (Monk & Bret J. Wagner, 2012). Juga menjabarkan bahwa terdapat tipe proses inti dan proses pendukung, proses inti merupakan alur aktivitas yang terjadi di dalam organisasi yang berkaitan langsung dengan organisasi tersebut. Sedangkan proses pendukung terjadi di luar lingkungan organisasi yang memberikan nilai secara tidak langsung sebagai contoh proses inti adalah proses produksi yang terjadi di dalam perusahaan. Sedangkan ekspedisi atau mengantar barang ke konsumen dilakukan oleh pihak ketiga merupakan contoh proses eksternal. Proses dapat juga menjadi alat untuk memodelkan dan melakukan analisis proses bisnis yang tengah berjalan di organisasi (*as-is*) dan proses bisnis usulan (*to-be*).

2.4.1 Analisis dan Pemodelan Proses Bisnis *As-Is*

Tahap ini merupakan aktivitas mengumpulkan data terkait kegiatan dan aktivitas bisnis, setelah data terkumpul data di analisis, setelah itu melakukan pemodelan berdasarkan hasil analisis dari pengumpulan data sehingga dapat menampilkan usulan tentang adanya *improvement* pada proses bisnis organisasi tersebut (Becker, Kugeler, & Rosemann, 2003).

Wawancara dengan pemangku kepentingan merupakan cara untuk mengetahui proses bisnis, atau melakukan observasi terhadap pihak-pihak yang melakukan aktivitas bisnis (Przybytek, 2011). Wawancara yang dilakukan pada penelitian ini menggunakan teknik wawancara semi-terstruktur, yang merupakan kategori *in-depth interview* yang pelaksanaannya lebih bebas bila daripada wawancara terstruktur. Tujuan wawancara jenis ini adalah untuk menemukan permasalahan secara lebih terbuka dengan pihak yang diajak wawancara. (Sugiyono, 2014). Przybytek juga mengemukakan bahwa setelah proses bisnis *as-is* teridentifikasi, hasil pemodelan itu harus memuat proses dasar dari organisasi tersebut.

2.4.2 Analisis dan Pemodelan Proses Bisnis *To-Be*

Merupakan kegiatan untuk menghasilkan pemodelan proses bisnis usulan yang dihasilkan dari analisis proses bisnis *as-is*, yang merupakan peningkatan proses bisnis sebelumnya (Becker et al., 2003), Becker juga mengemukakan bahwa untuk proses pemindahan sistem dapat dieksekusi dengan rentang waktu menengah atau panjang. Penerapan otomatisasi belum tentu menjamin peningkatan, oleh karena keikutsertaan pemangku kepentingan dalam tahap ini sangat diharuskan (Przybytek, 2011).

2.5 Analisis Persyaratan

Pada tahap ini peneliti menggunakan model *use case*, model ini terbukti meningkatkan pemahaman dan menjadikan pengertian tentang persyaratan sistem tidak ambigu, dan secara tidak langsung meningkatkan waktu pengembangan (Bittner & Spence, 2002). Pemodelan ini berguna untuk mengenali pemangku kepentingan (pihak yang terlibat merasakan hasil dari sistem), lalu kebutuhan pemangku kepentingan, hingga fitur dan persyaratan dari sistem.

2.5.1 Identifikasi Tipe Pemangku Kepentingan

Merupakan pihak-pihak yang merasakan hasil dari pengembangan sistem ini nantinya, identifikasi untuk tipe *stakeholder* terdiri dari (Bittner & Spence, 2002):

1. Pengguna (*Users*), merupakan pemangku kepentingan yang secara langsung menggunakan dan merasakan manfaat langsung dari sistem *use case*;
2. Sponsor (*Sponsors*), tipe pemangku kepentingan ini mungkin tidak mendapatkan manfaat secara langsung, tetapi pihak ini yang membiayai pengeluaran yang dilakukan untuk pengembangan sistem ini, sangat penting untuk mempertimbangkan usulan mereka;
3. Pengembang (*Developers*), adalah pemangku kepentingan yang secara langsung menghasilkan produk sistem yang dikembangkan;
4. Pihak yang berwenang (*Authorities*), adalah pihak-pihak yang mengatur tentang aturan dan regulasi pada organisasi;
5. *Customers* (Pelanggan), adalah pihak yang akan membeli sistem dan merasakan manfaat secara langsung.

2.5.2 Analisis Masalah

Tahap ini merupakan aktivitas untuk menganalisis masalah dan juga memberikan saran/solusi untuk memenuhi kebutuhan pengguna (Leffingwell & Widrig, 2003). Diharapkan dengan tahap ini dapat diperoleh pemahaman yang lebih baik akan masalah. Praktik terbaik untuk menganalisis masalah adalah menggunakan tabulasi dengan tabel pernyataan masalah atau di dalam buku (Bittner & Spence, 2002) biasa disebut *problem statement*. Tabel di bawah ini merupakan contoh tabel yang nanti akan digunakan ketika analisis masalah.

Tabel 2.4 Kerangka Dokumentasi Pernyataan Masalah

<i>The Problem Of</i>	Berisi deskripsi singkat dan jelas dari masalah.
<i>Affects</i>	Berisi <i>stakeholder</i> yang dipengaruhi oleh masalah di atas.
<i>The Impact of which is</i>	Pada bagian berisi dampak masalah terhadap <i>stakeholder</i> dan kegiatan bisnis dengan deskripsi.
<i>A succesful solution would</i>	Pada bagian ini adalah deskripsi yang mengandung solusi yang ditawarkan.

Sumber : (Bittner & Spence, 2002)

2.5.3 Identifikasi Kebutuhan Pemangku Kepentingan

Tahap ini merupakan kelanjutan setelah masalah sudah jelas dipahami, tahap ini berisi deskripsi singkat yang dapat mewakili masalah yang dikenali, bisa saja mengandung masalah bisnis, personal atau operasional, yang harus diperhatikan adalah pertimbangan untuk menggunakan sistem yang baru (Leffingwell & Widrig, 2003).

Tahap ini merupakan tahap yang menunjukkan kebutuhan pengguna lebih jelas karena dideskripsikan dengan pernyataan-pernyataan singkat yang disampaikan oleh para pemangku kepentingan (Bittner & Spence, 2002). Untuk mengidentifikasi kebutuhan bisa menggunakan praktik berikut: Wawancara, *Requirement workshop*, *Brainstorming* and *Idea Reduction*, *Storyboarding* (Leffingwell & Widrig, 2003).

2.5.4 Identifikasi Fitur

Tahap ini merupakan tahap untuk mengenali solusi yang ditawarkan sistem untuk memenuhi satu atau lebih kebutuhan *stakeholder* yang disebut sebagai fitur (Leffingwell & Widrig, 2003), fitur mendeskripsikan rangkuman manfaat yang ditawarkan oleh sistem (produk) nantinya (Bittner & Spence, 2002).

Tabel 2.5 Kerangka Tabel fitur

Kode Fitur	Deskripsi
Berisi kode identitas fitur	Berisikan deskripsi singkat fitur dari sistem

Diadaptasi dari : (Bittner & Spence, 2002)

2.5.5 Identifikasi Persyaratan Fungsional dan Nonfungsional

Merupakan pernyataan kesepakatan persyaratan apa saja yang akan dikembangkan di sistem yang berasal dari kebutuhan pengguna dan fitur (Bittner & Spence, 2002). Persyaratan sistem ini akan ditampilkan dengan kalimat deklaratif, persyaratan fungsional merupakan reaksi sistem jika dimasukkan parameter tertentu dan dalam kondisi tertentu (Sommerville, 2010). Sedangkan persyaratan nonfungsional merupakan kualitas lebih dari sistem atau *constraint*

tertentu, contoh persyaratan nonfungsional antara lain *performance*, *usability*, *reliability*, sehingga analisis persyaratan sangat penting.

2.5.6 Pemodelan Use Case

Diagram use case tidak berisi penjelasan atau cerita tentang use case sehingga dibutuhkan penjelasan lebih lanjut tentang persyaratan fungsional sistem, tabel di bawah merupakan format spesifikasi yang berisi cerita dan penjelasan bagaimana aktor berinteraksi yang digambarkan melalui diagram use case.

Tabel 2.6 Format Spesifikasi Use Case

Brief Description	Berisi penjelasan singkat dari <i>use case</i>
Actor	Berisi aktor yang berkaitan dengan <i>use case</i>
Pre-Condition	Berisi satu atau lebih kondisi yang harus dipenuhi oleh sistem sebelum menjalankan <i>use case</i>
Post-Condition	Berisi satu atau lebih kondisi yang dipenuhi oleh sistem setelah <i>use case</i> berakhir
Basic Flow	Berisi alur <i>basic</i> yang dilalui oleh <i>use case</i> dari awal hingga akhir jika berjalan normal
Alternative Flow	Berisi alur opsional ketika <i>use case</i> tidak berjalan seperti yang diharapkan
Subflow	Berisi satu atau lebih kelompok alur yang merupakan bentuk sederhana dari <i>use case</i> yang detail.

Sumber : Diadaptasi dari (Bittner & Spence, 2002)

2.6 Unified Modelling Language (UML)

Merupakan standar dalam pemodelan dan perancangan perangkat lunak, yang mampu mendokumentasikan artefak yang diperlukan dalam pengembangan sistem (Pressman, 2010). UML berguna untuk memberikan pemahaman yang tidak ambigu ketika proses pengembangan sistem.

Ada dua jenis atau tipe dari diagram UML antara lain diagram struktur yang menampilkan struktur sistem secara statis dan diagram *behaviour* dimana mewakili diagram dinamis yang bersifat berubah-ubah, berikut merupakan tipe diagram struktur:

1. *Class diagram*
2. *Profile diagram*
3. *Composite structure diagram*
4. *Component diagram*
5. *Deployment diagram*

6. *Object diagram*
7. *Package diagram*







Diagram *behaviour* menampilkan pemodelan yang bersifat dinamis (berubah-ubah) yang terpengaruh oleh satuan waktu dan pertukaran pesan antar entitasnya, dibawah ini adalah diagram yang jenisnya *behaviour*:

1. *Activity diagram*
2. *Use case diagram*
3. *State machine diagram*
4. Interaction diagram, jenis interaction diagram antara lain:
 - *Sequence diagram*
 - *Communication diagram*
 - *Interaction overview diagram*
 - *Timing diagram*




2.6.1 Use-Case Diagram

Diagram ini menunjukkan tingkah laku (*behaviour*) dari sistem yang merupakan kumpulan fungsionalitas yang sistem sediakan kepada pengguna, dengan diagram ini juga dapat melihat bagaimana interaksi aktor dengan sistem.(Booch et al., 2007).

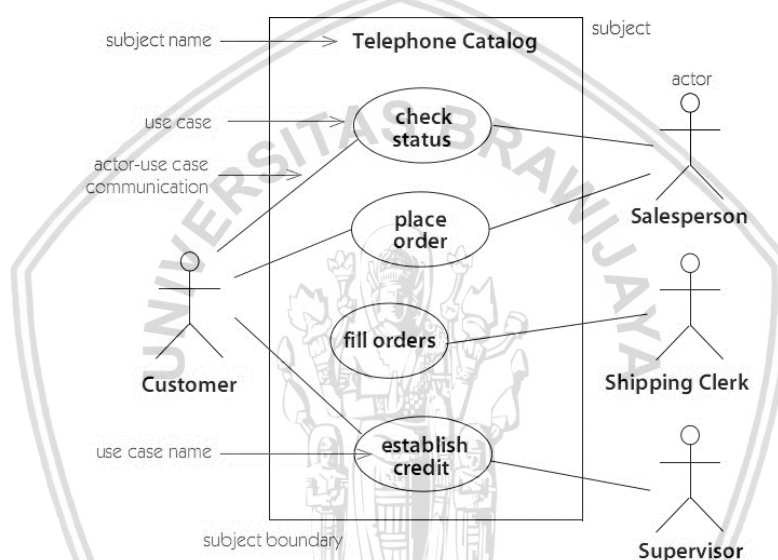
Tabel 2.7 Simbol Use Case Diagram

No.	Simbol	Nama	Deskripsi
1		<i>Use Case</i>	Deskripsi dari <i>use case</i> berisi kata kerja di depan yang mengisyaratkan mengerjakan sesuatu
2		<i>Actor</i>	Menyepesifikasikan himpunan peran dari pengguna yang berinteraksi dengan sistem melalui <i>use case</i>
3		<i>Association</i>	Menggambarkan relasi antara objek satu dengan objek lainnya.
4		<i>Extend</i>	Menyepesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan
5		<i>Include</i>	Menyepesifikasikan bahwa <i>use case</i> sumber secara eksplisit, tetapi tidak menggambarkan urutan
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung pada elemen yang tidak mandiri

Tabel 2.7 Simbol Use Case Diagram (Lanjutan)

7		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) mendapatkan perilaku dan struktur data dari objek yang ada di atasnya yaitu objek induk
8		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi)
9		<i>System</i>	Menyepifikasikan batas sistem

Di bawah ini adalah contoh diagram use case :




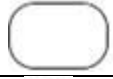

Gambar 2.14 Contoh use case diagram



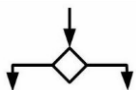
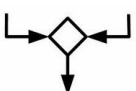
Sumber: (Booch, Rumbaugh, & Jacobson, 2005)

2.6.2 Activity Diagram

Diagram yang menggambarkan alur yang mewakili proses dari sebuah sistem, seperti namanya diagram ini berfokus pada aktivitas dan tanggung jawab dari proses tersebut

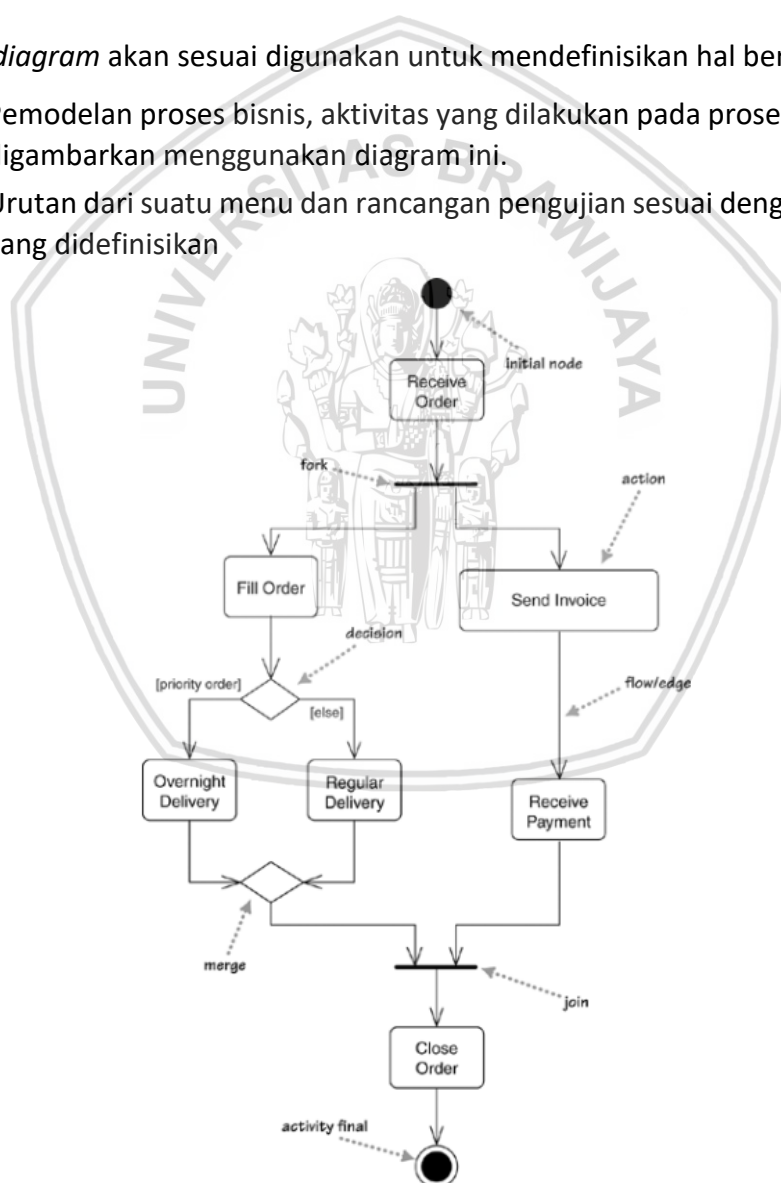
Tabel 2.8 Simbol Activity Diagram

No.	Simbol	Nama	Deskripsi
1		<i>Activity</i>	Memperlihatkan bagaimana interaksi masing-masing <i>interface</i> satu dengan yang lainnya
2		<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali

4		<i>Activity final node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>fork node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran atau sebaliknya dari beberapa aliran ke satu aliran
6		<i>Decision node</i>	Seleksi kondisi dari satu aliran menjadi dua sampai lebih aliran dan memilih salah satu aliran
7		<i>Merge node</i>	Merupakan kembalian dari seleksi kondisi, aliran yang masuk harus sama dengan aliran yang keluar dari <i>Decision node</i>

Activity diagram akan sesuai digunakan untuk mendefinisikan hal berikut :

1. Pemodelan proses bisnis, aktivitas yang dilakukan pada proses bisnis bisa digambarkan menggunakan diagram ini.
2. Urutan dari suatu menu dan rancangan pengujian sesuai dengan kasus uji yang didefinisikan



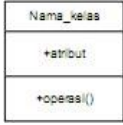
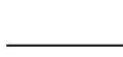



Gambar 2.15 Contoh Activity Diagram

Sumber : (Fowler, 2004)

2.6.3 Class Diagram

Berisi struktur dari suatu sistem yang mewakili atribut, operasi dan hubungan antar kelas lain (Booch et al., 2007), tabel di bawah ini berisi penjelasan dan deskripsi singkat.

Tabel 2.9 Simbol *Class Diagram*

No.	Simbol	Nama	Deskripsi
1		<i>Class</i>	<i>Class</i> merupakan cetakan dari objek yang merupakan komponen sistem. <i>Class</i> mendefinisikan atribut, operasi dan <i>instance</i> .
2		<i>Association</i>	Relasi yang menunjukkan hubungan antara dua <i>class</i> .
3		<i>Generalization</i>	Relasi yang menunjukkan pewarisan atribut dan operasi-operasinya dari <i>class</i> induk(<i>parent</i>) ke <i>class</i> anak (<i>child</i>).
4		<i>Dependency</i>	Relasi yang menunjukkan suatu <i>class</i> ketergantungan pada <i>class</i> yang lain.
5		<i>Aggregation</i>	Relasi yang menunjukkan sebuah elemen yang terdiri dari beberapa komponen kecil. Biasanya dilengkapi dengan atribut tambahan seperti keterangan (1..* dsb.)

Ada tiga tingkat pemodelan *class diagram* berdasarkan tingkat perinciannya sesuai dalam tahap-tahap pengembangan suatu sistem (Kendall & Apress, 2004)

1. *Domain - Level Class Diagram*

Kegunaan utama level ini adalah penamaan pada *class* dengan istilah umum karena pada tahap ini belum tampak operasi atau atribut yang digunakan oleh *class*

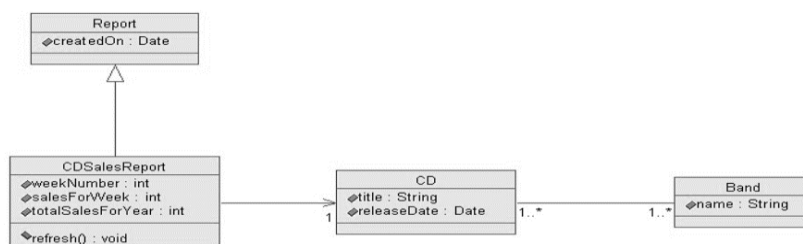
2. *Analysis - Level Class Diagram*

Pada tingkat ini *class* pada level sebelumnya ditambahkan atribut yang ditambahkan setelah memasuki fase analisis, tanpa menambahkan operasi

3. *Design - Level Class Diagram*

Level ini merupakan tahap dimana semua atribut dan operasi dimasukkan pada *class* lalu digunakan semua notasi standar UML untuk menggambarkan perancangan sistem nantinya

Gambar di bawah ini merupakan contoh dari *class diagram*



Gambar 2.16 Contoh Class Diagram

Sumber: (Bell, 2003)

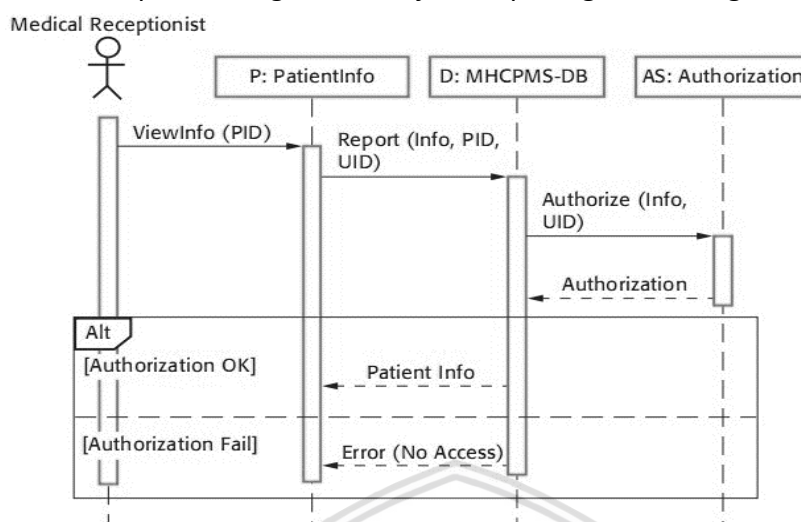
2.6.4 Sequence Diagram

Dengan diagram ini maka dapat mengetahui gambaran urutan (*sequence*) antar objek yang berinteraksi dengan aktor, sangat membantu untuk perancangan yang lebih mendetail karena diagram ini menampilkan pesan apa yang dikirimkan antar entitas berdasarkan satuan waktu pada suatu skenario dalam *use case* (Sommerville, 2010).

Tabel 2.10 Simbol Sequence Diagram

No.	Simbol	Nama	Deskripsi
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi
2		<i>Actor</i>	Merupakan aktor pada diagram ini, berperan untuk memicu atau memulai
3		<i>Boundary</i>	Merupakan <i>interface</i> aktor berinteraksi
4		<i>Control</i>	Merupakan logika atau proses dari suatu <i>sequence</i>
5		<i>Model</i>	Merupakan domain atau letak data berada
6		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi
7		<i>Reply Message</i>	Pesan kembalian setelah memanggil, hanya untuk panggilan yang memiliki nilai kembalian
8		<i>Lost Message</i>	Pesan yang tidak diketahui kembaliannya dan di luar lingkup dari <i>sequence</i> tersebut

Contoh *sequence diagram* ditunjukkan pada gambar diagram di bawah ini



Gambar 2.17 Contoh Sequence Diagram

Sumber: (Sommerville, 2010)

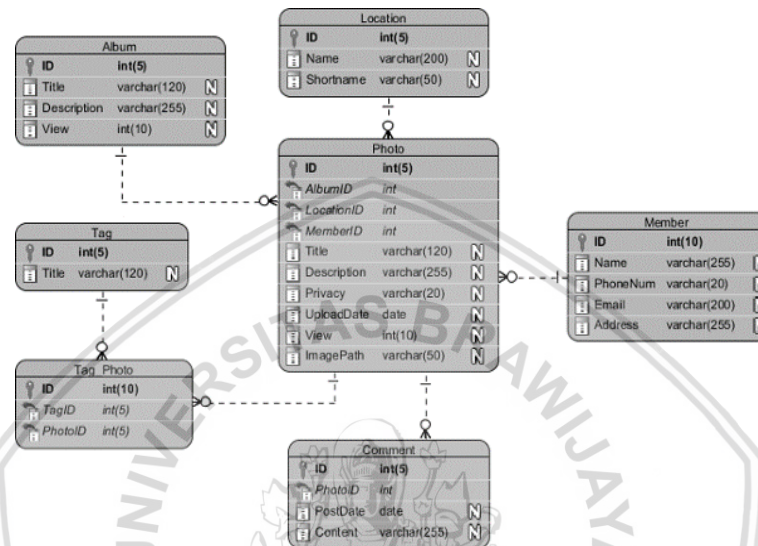
2.7 Physical Data Model (PDM)

Physical Data Model merepresentasikan desain sebenarnya dari *database* relasional. Ini mewakili bagaimana data harus terstruktur dan terkait dalam DBMS tertentu sehingga penting untuk mempertimbangkan konvensi dan pembatasan DBMS saat merancang ERD fisik. Ini berarti bahwa penggunaan tipe data yang akurat diperlukan untuk kolom entitas dan penggunaan kata-kata yang dicadangkan harus dihindari dalam memberi nama entitas dan kolom. Selain itu, desainer *database* juga dapat menambahkan *primary key*, *foreign key* dan *constraints* pada desain. (VisualParadigm, n.d.)

2.7.1 Karakteristik umum dari *Physical Data Model*:

- Biasanya menggambarkan persyaratan data untuk satu proyek atau aplikasi. Terkadang bahkan sebagian aplikasi;
- Dapat diintegrasikan dengan *Physical Data Model* lainnya melalui *repository of shared entities*;
- Biasanya berisi 10-1000 tabel, meskipun angka ini sangat bervariasi tergantung pada ruang lingkup model data;
- Berisi hubungan antar tabel yang menangani kardinalitas dan *nullability* (*optionality*) dari hubungan;
- Dirancang dan dikembangkan agar bergantung pada versi spesifik DBMS, lokasi penyimpanan data atau teknologi;
- Kolom akan memiliki tipe data dengan presisi dan panjang yang ditetapkan;
- Kolom akan memiliki kelayakan (opsional) yang ditetapkan;

- Tabel dan kolom akan memiliki definisi;
- Juga akan mencakup objek fisik lainnya seperti *views*, batasan *primary key*, batasan *foreign key*, indeks, *security roles*, *store procedures*, ekstensi XML, penyimpanan file, dll;
- *Physical Data Model* mungkin hanya menunjukkan sebagian kecil data meta yang terdapat dalam model. ("What are Conceptual, Logical and Physical Data Models? |," n.d.).

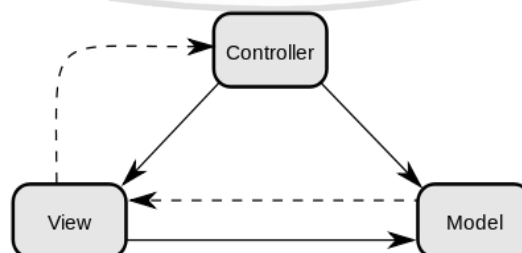


Gambar 2.18 Contoh *Physical Data Model*

Sumber : (VisualParadigm, n.d.)

2.8 Model View Controller (MVC)

Merupakan konsep atau bisa dibilang *pattern* untuk memisahkan *source code* dari entitas yang lain antara lain tampilan (*view*) pusat logika (*controller*) dan model (yang mengurus tentang data), dengan *pattern* ini dapat mengurangi kebingungan dalam melakukan pengembangan hingga nantinya pemeliharaan sistem. (Vuksanovic & Sudarevic, 2011)



Gambar 2.19 Konsep *Model-view-controller*

MVC sendiri terdiri dari 3 bagian, antara lain:

a. Model

Model merepresentasikan struktur data yang dibangun. Biasanya terdiri atas fungsi terhadap *database*.

b. *View*

Berisi informasi dan tampilan, kode di bagian inilah yang akan ditampilkan pada pengguna, tidak hanya tampilan antarmuka, formulir juga bisa dikategorikan sebagai *view*.

c. *Controller*

Di bagian ini pusat logika atau bisa dikatakan yang mengatur *view* hingga model semua diatur pada bagian ini.

2.9 Analisis Inkremental & Iterasi

Analisis inkremental adalah membagi sistem menjadi beberapa bagian untuk dibangun, banyak teknik pembagian yang bisa digunakan tergantung kebutuhan dari sistem, selanjutnya membuat perkiraan dari setiap inkremen berapa kali untuk melakukan iterasi (periode). (Cockburn, 2008).

2.10 Pengujian

Pengujian merupakan tahapan dimana *software*/sistem diuji apakah sistem berjalan seperti yang diharapkan/direncanakan, tujuannya untuk mencari kesalahan dan solusi yang terbaik untuk sistem tersebut, pengujian bisa mencakup fungsional (persyaratan fungsional sistem) atau kualitas, *constraint* atau keadaan tertentu seperti *usability*, performa, atau reliabilitas sistem yang biasa disebut dengan pengujian nonfungsional (Agarwal, Tayal, & Gupta, 2010)

2.10.1 Scenario Testing

Skenario adalah kisah yang menggambarkan situasi hipotetis. Dalam pengujian, Anda memeriksa bagaimana program berupaya dengan situasi hipotetis ini. Tes skenario yang ideal adalah kredibel, memotivasi, mudah dievaluasi, dan kompleks. Pesan utama dari pola ini adalah Anda harus menjaga keempat atribut ini diingat ketika Anda merancang tes skenario dan berusaha keras untuk mencapainya.

Variasi penting dari tes skenario melibatkan tes yang lebih keras. Ceritanya akan sering melibatkan urutan, atau nilai data, yang jarang digunakan oleh pengguna biasa. Mereka mungkin muncul, bagaimanapun, keluar dari kesalahan pengguna atau dalam perjalanan situasi yang tidak biasa tetapi masuk akal, atau dalam perilaku pengguna yang tidak bersahabat. Skenario semacam itu biasa terjadi dalam pengujian keamanan atau bentuk lain dari *stress testing*.

Skenario menentukan aktor, peran, proses bisnis, tujuan dari aktor, dan peristiwa yang dapat terjadi dalam upaya mencapai tujuan. Skenario adalah sebuah Instansiasi kasus penggunaan. Skenario sederhana menelusuri satu kasus penggunaan tunggal, menentukan datanya nilai-nilai dan dengan demikian jalan diambil melalui kasus ini. Kasus penggunaan yang lebih kompleks melibatkan

Rangkaian beberapa kasus penggunaan, untuk melacak tugas yang diberikan, dari ujung ke ujung. (Kaner, 2003b)

- Beberapa kelompok membuat kumpulan tes skenario sebagai tes regresi.
- Orang lain menjalankan skenario sekali atau beberapa kali dan kemudian mendesain skenario yang lain.

Penguji sering mengembangkan skenario untuk mengembangkan wawasan ke dalam produk. Ini terutama benar sejak awal dalam pengujian dan pengujian lagi akhir (ketika produk telah stabil dan tester sedang mencoba memahami kegunaan lanjutan dari produk.)

2.10.2 Pengujian *Usability* menggunakan *USE QUESTIONNAIRE*

Pengujian yang dilakukan terhadap pengguna untuk mengetahui tingkat kegunaan (*usability*) dari suatu sistem. Pengujian dapat menggunakan kuesioner atau observasi langsung terhadap sistem atau yang lainnya, pengujian bisa dilakukan ketika produk dikembangkan atau setelah digunakan beberapa lama atau pada tahap eksplorasi (Churm, 2012). Di bawah ini merupakan parameter untuk mengukur *usability* :

a. Mudah dipelajari (*easy to use*)

Kualitas sistem termasuk antarmuka yang menunjukkan apakah sistem mudah untuk dipelajari dan digunakan dalam menyelesaikan tugas tertentu.

b. Efisiensi (*efficiency*)

Dengan *effort* yang sedikit apakah mampu menghasilkan hasil yang bagus dan sesuai harapan.

c. Mudah diingat (*memorability*)

Kemampuan sistem untuk mudah diingat, baik dari sisi fitur atau menu-menu yang ada maupun cara menggunakannya secara maksimal

d. Kesalahan dan keamanan (*errors*)

Perlindungan dan pertolongan kepada pengguna terhadap kondisi dan situasi yang tidak diinginkan dan berbahaya ketika mengoperasikan sistem, contohnya antara lain *menu help* untuk memberi solusi, dan konfirmasi penghapusan berkas.

e. Kepuasan (*satisfaction*)

Menunjuk kepada suatu keadaan dimana pengguna merasa puas setelah menggunakan sistem tersebut karena kemudahan yang dimiliki oleh sistem. Semakin pengguna menyukai suatu sistem, secara implisit mereka merasa puas dengan sistem yang dimaksud.

Menguji *usability* berarti menguji tingkat efektif, efisien, kepuasan yang dirasakan oleh pengguna, ada beberapa metrik atau ukuran untuk menguji *usability*, metrik tersebut biasanya sebuah kuesioner yang akan diisi oleh pengguna yang telah memakai sistem/ aplikasi.

USE QUESTIONNAIRE merupakan salah satu metrik untuk mengukur tingkat *usability* dari *user interface* sebuah sistem/aplikasi, metrik ini menggunakan tiga dimensi yaitu *usefulness*, *ease of use*, dan *satisfaction*, dan untuk faktor yang memengaruhi *ease of use* dapat dibagi menjadi 2 yaitu *ease of learning* dan *ease of use*. (Lund, 2001)

Berikut merupakan paket kuesioner USE :

Usefulness

1. *It helps me be more effective.*
2. *It helps me be more productive.*
3. *It is useful.*
4. *It gives me more control over the activities in my life.*
5. *It makes the things I want to accomplish easier to get done.*
6. *It saves me time when I use it.*
7. *It meets my needs.*
8. *It does everything I would expect it to do.*

Ease of Use

9. *It is easy to use.*
10. *It is simple to use.*
11. *It is user friendly.*
12. *It requires the fewest steps possible to accomplish what I want to do with it.*
13. *It is flexible.*
14. *Using it is effortless.*
15. *I can use it without written instructions.*
16. *I don't notice any inconsistencies as I use it.*
17. *Both occasional and regular users would like it.*
18. *I can recover from mistakes quickly and easily.*
19. *I can use it successfully every time.*

Ease of Learning

20. *I learned to use it quickly.*
21. *I easily remember how to use it.*
22. *It is easy to learn to use it.*
23. *I quickly became skillful with it.*

Satisfaction

24. *I am satisfied with it.*
25. *I would recommend it to a friend.*
26. *It is fun to use.*
27. *It works the way I want it to work.*
28. *It is wonderful.*
29. *I feel I need to have it.*
30. *It is pleasant to use.*

2.10.3 Skala *Likert*

Dalam penelitian ini skala *likert* digunakan untuk mengukur seberapa besar tingkat *usability* dari sistem menggunakan kuesioner yang diisi oleh responden. Skala *likert* adalah penentuan skala dalam pemberian respons dalam jawaban (Risnita, 2012), jawaban dari responden mempunyai 5 tingkat seperti di bawah ini:

- a. Sangat setuju
- b. Setuju
- c. Netral
- d. Tidak setuju
- e. Sangat tidak setuju

Analisis kuantitatif membutuhkan skor pada tiap jawaban yang ditunjukkan pada Tabel 2.11.

Tabel 2.11 Penilaian jawaban kuantitatif

Jawaban	Skor
Sangat setuju	5
Setuju	4
Netral	3
Tidak setuju	2
Sangat tidak setuju	1

Skala *Likert* diterapkan dengan cara menghitung agar didapatkan indeks persentase pengujian *usability*. Persamaan 2.1 adalah rumus untuk menghitung indeks dari tiap soal kuesioner yang diberikan, *nilaiSTS* adalah banyak nilai dari jawaban bernilai 1 (sangat tidak setuju), *nilaiTS* merupakan banyak nilai dari jawaban bernilai 2 (tidak setuju), *nilaiN* merupakan banyak nilai dari jawaban bernilai 3 (netral), *nilaiS* merupakan banyak nilai yang diberikan bagi jawaban bernilai 4 (Setuju) dan *nilaiSS* merupakan banyak nilai yang diberikan bagi jawaban bernilai 5 (sangat setuju).

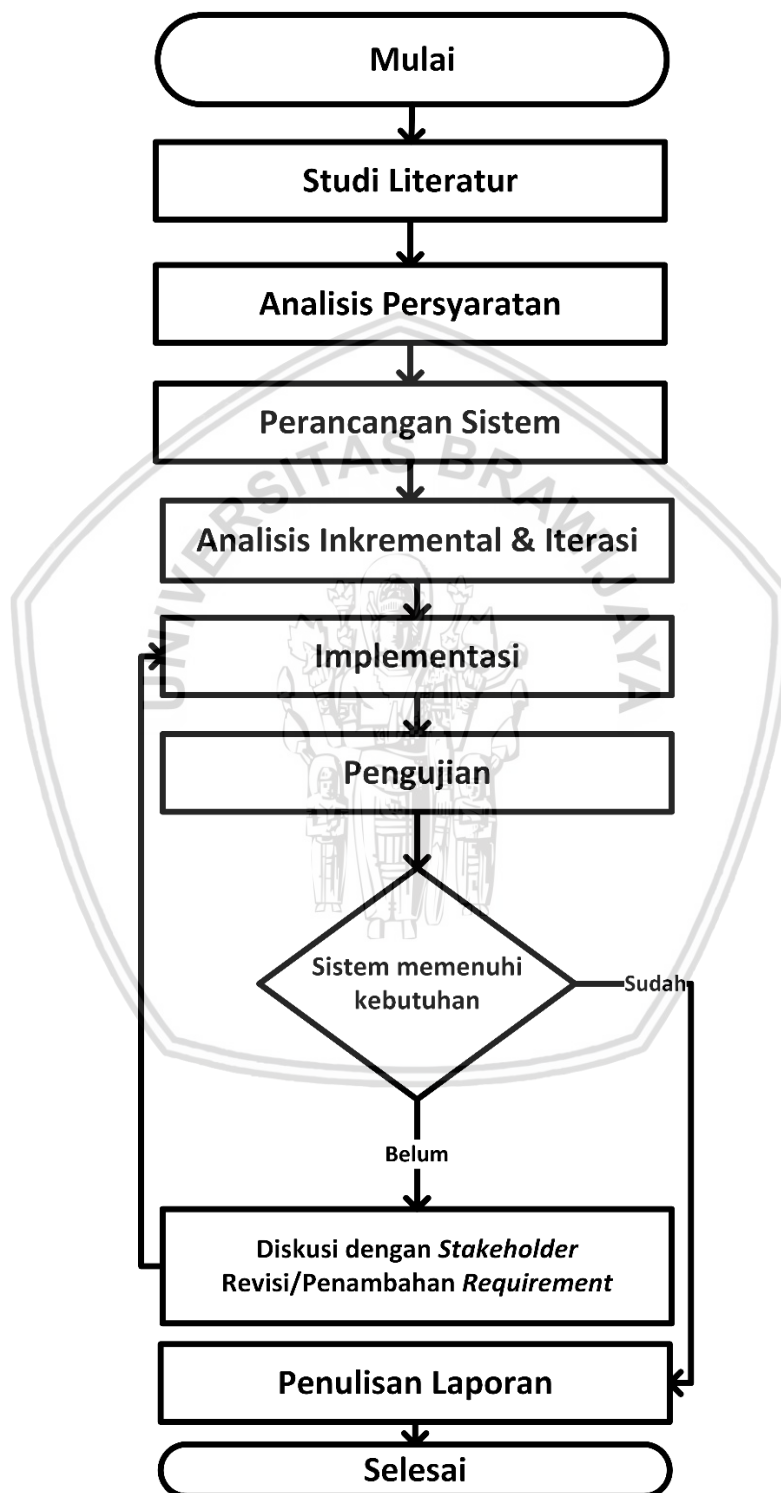
Semua nilai berikut dijumlahkan lalu dibagi dengan *skorLikertTertinggi* yang merupakan skor tertinggi pada kuesioner ini yaitu 5 yang dikali dengan *jumlahResponden* yang merupakan banyaknya responden yang mengisi kuesioner ini, lalu semuanya dikali dengan 100% untuk menunjukkan nilai persentase.

Persamaan 2.1 Rumus menghitung indeks pada skala *likert*

$$\text{Index (\%)} = \frac{(\text{nilaiSTS} \times 1) + (\text{nilaiTS} \times 2) + (\text{nilaiN} \times 3) + (\text{nilaiS} \times 4) + (\text{nilaiSS} \times 5)}{\text{SkorLikertTertinggi} \times \text{JumlahResponden}} \times 100\%$$

BAB 3 METODOLOGI

Tipe penelitian ini merupakan implementatif, berikut diagram alir untuk menjelaskan tahapan dalam yang dilakukan dalam penelitian ini



Gambar 3.1 Diagram Metodologi Penelitian
Adaptasi dari : (Cockburn, 2008)

3.1 Studi Literatur

Tahap ini penulis mencari referensi dan literatur seperti buku atau *e-book*, jurnal hingga tesis, skripsi dan didapatkan dari hasil pencarian di internet sehingga dapat mendukung penelitian ini

3.2 Analisis Persyaratan

Analisis persyaratan bertujuan untuk mengembangkan model yang menggambarkan ketika sistem berjalan memenuhi kebutuhan pemangku kepentingan, dalam penelitian ini adalah seksi rehabilitasi BNN Kota Malang

Aktivitas yang akan dilakukan:

1. Membuat daftar pertanyaan wawancara;
2. Melakukan wawancara kepada pegawai seksi rehabilitasi;
3. Meminta *Standart Of Procedure* (SOP) yang berlaku di seksi rehabilitasi;
4. Dari berbagai skenario/prosedur, memilih aktivitas yang akan di fasilitasi oleh sistem;
5. Mengelola hasil wawancara menjadi daftar kebutuhan sistem;
6. Berdasarkan hasil wawancara, membuat pemodelan proses bisnis sistem yang sudah ada;
7. Membuat *use case diagram*
 - a. Mengidentifikasi aktor
 - b. Mengidentifikasi kebutuhan tiap aktor
 - c. Mengidentifikasi batasan sistem
8. Membuat pemodelan proses bisnis usulan;
9. Membuat *Activity Diagram*.

Pada tahap ini menghasilkan proses bisnis to-be, tabel daftar kebutuhan pengguna, tabel fitur, *use case diagram*, spesifikasi *use case*, *activity diagram* beserta tabel-tabel untuk melihat hubungan semuanya.

3.3 Perancangan Sistem

Perancangan sistem bertujuan untuk mentransformasi model analisis menjadi model perancangan yang menjadi *blueprint* untuk mengembangkan sistem informasi data pasien. Masukan dari tahap ini adalah daftar kebutuhan sistem, bisnis proses yang sedang berjalan.

Aktivitas yang akan dilakukan:

1. Membuat *Sequence Diagram*;
2. Membuat *Class Diagram*;
3. Membuat *Physical Data Model*;

4. Membuat *prototype*;
5. Membuat *test case* (Skenario).

Tahap ini menghasilkan *blueprint* sistem (*sequence diagram*, *class diagram*, *physical data model*, *prototype user interface*).

3.4 Analisis Inkremental & Iterasi

Tahap ini bertujuan untuk memisahkan bagian apa saja yang akan dikerjakan ulang (*iterative*) dan bagian-bagian apa yang akan dikerjakan secara bertahap (*increment*) (Cockburn, 2008). Dalam penelitian ini tahap ini dilakukan agar peneliti dapat membagi dan mengerjakan bagian-bagian dari sistem informasi pasien rehabilitasi BNN Kota Malang. Daftar kebutuhan sistem, bisnis proses yang sedang berjalan adalah informasi yang dibutuhkan agar tahap ini dapat berjalan.

Aktivitas yang akan dilakukan:

1. Mengidentifikasi kebutuhan yang harus terpenuhi dahulu;
2. Mengurutkan pengerjaan tiap kebutuhan.

Analisis Inkremental & Iterasi menghasilkan jadwal *milestone* dari implementasi sistem yang berisi petunjuk bagian mana yang harus dikerjakan ulang dan bagian yang dikerjakan bertahap.

3.5 Implementasi

Pada tahap ini penulis mengembangkan sistem, mentransformasi rancangan menjadi sistem yang berjalan sesuai kebutuhan dari sistem informasi pasien rehabilitasi BNN Kota Malang. Masukan dari tahap ini hampir dari tahap-tahap sebelumnya yaitu *use case diagram*, pemodelan proses bisnis usulan, *Activity diagram*, *Class diagram*, *PDM*, *Sequence diagram*, *prototype user interface*, dan urutan implementasi (*milestone*) yang didapatkan dari tahap analisis *iterative & incremental*.

Aktivitas yang akan dilakukan:

1. Jika ada revisi, memperbaiki sistem sesuai revisi yang disepakati bersama;
2. Implementasi kebutuhan sesuai urutan dari analisis *incremental*.

Pada tahap ini menghasilkan sistem yang bisa diuji yang diimplementasi sesuai dari rancangan yang sudah diberikan dan selanjutnya akan dilakukan pengujian sistem.

3.6 Pengujian

Pengujian dilakukan untuk mengetahui apakah sistem dikembangkan sesuai dengan persyaratan yang ditentukan pada tahap analisis persyaratan, dan jika terjadi kesalahan dapat ditemukan secepatnya untuk diperbaiki. Adapun masukan dari tahap ini antara lain, daftar persyaratan sistem

(fungsional/nonfungsional), sistem yang bisa diuji (artefak hasil implementasi), serta proses bisnis *to-be*.

Aktivitas yang akan dilakukan:

1. Menguji Sistem menggunakan *test case* (demo di depan pemangku kepentingan);
2. Menganalisis hasil pengujian.

Tahap pengujian menghasilkan dokumentasi pengujian yang berguna untuk melakukan perbaikan sistem jika terdapat beberapa kesalahan atau tidak disetujui oleh pemangku kepentingan.

3.7 Diskusi dengan pemangku kepentingan

Tahap ini merupakan kelanjutan dari pengujian yaitu, menemukan solusi bersama jika terdapat kesalahan pada sistem yang dikembangkan sehingga menghasilkan iterasi selanjutnya untuk melakukan peningkatan/perbaikan (Cockburn, 2008). Dalam penelitian ini peneliti melakukan diskusi setelah sistem menjalani pengujian, dan dengan dokumentasi pengujian melakukan verifikasi apakah sistem dikembangkan sesuai dengan kebutuhan organisasi. Masukan dari tahap ini adalah dokumentasi pengujian beserta proses bisnis usulan.

Aktivitas yang akan dilakukan:

1. Berdiskusi jika hasil testing tidak seperti yang diharapkan;
2. Menampung hasil revisi.

Tahap ini menghasilkan dokumentasi diskusi yang berisi bagian-bagian apakah dari sistem yang perlu diperbaiki, penyempurnaan fitur atau menambahkan fitur.

3.8 Penulisan Laporan

Penulisan laporan bisa berjalan sementara pengembangan sistem berjalan, kecuali dalam bab kesimpulan (berisi hasil pengujian dan analisis) yang hanya bisa ditulis setelah semua tahapan selesai dan menghasilkan suatu hasil, yang mungkin masih belum sempurna dan mempunyai kekurangan yang nantinya bisa menjadi referensi untuk penelitian selanjutnya.

BAB 4 ANALISIS PERSYARATAN

Bab ini terdiri dari hasil analisis persyaratan, data dikumpulkan berdasarkan wawancara dengan pemangku kepentingan, para staf Rehabilitasi BNN Kota Malang, analisis dokumen standar prosedur dan juga observasi, hasil dari aktivitas analisis persyaratan adalah model proses bisnis yang tengah berjalan dan usulan model proses bisnis masa depan, tipe pemangku kepentingan, pernyataan masalah, kebutuhan pengguna, tipe pengguna sistem informasi nantinya, fitur yang ditawarkan oleh sistem informasi, persyaratan fungsional dan non-fungsional sistem, *use case* dan pemodelan aktivitas menggunakan *activity diagram*.

4.1 Pemodelan Proses Bisnis

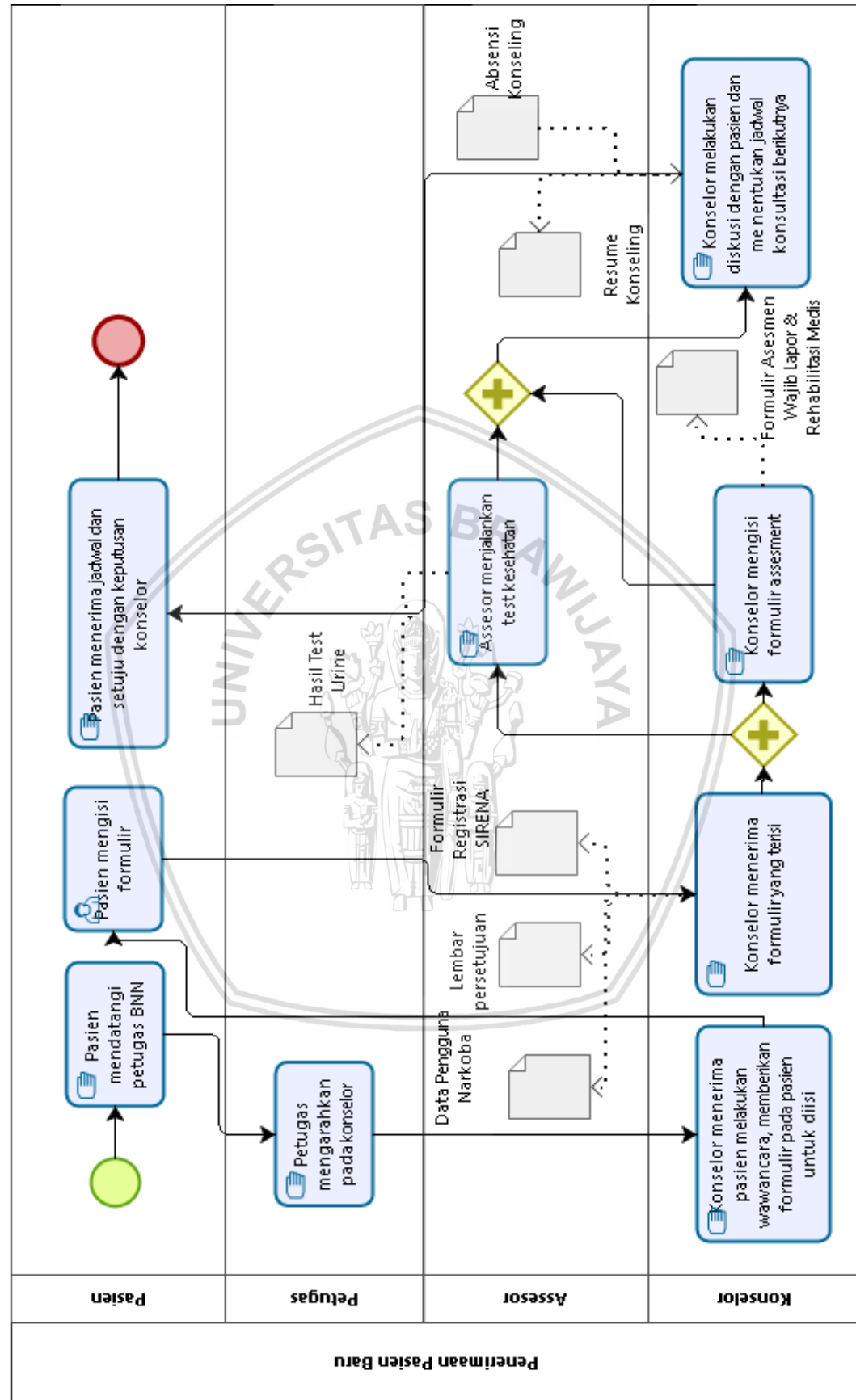
Pemodelan proses bisnis berguna untuk memberikan gambaran aktivitas apa saja yang tengah berlangsung untuk memenuhi kebutuhan dari seksi rehabilitasi, sehingga dapat memberikan gambaran tentang masalah dan usulan peningkatan dalam proses bisnis nantinya. Pengumpulan data dilakukan pada empat informan dengan teknik wawancara tidak terstruktur. Wawancara dilakukan secara langsung di BNN Kota Malang selama dua hari, hasil wawancara dapat dilihat di LAMPIRAN A.

4.1.1 Proses Bisnis *As-Is*

Proses bisnis *as-is* merupakan keadaan bisnis yang tengah berjalan, digunakan oleh peneliti untuk mengidentifikasi masalah yang ada pada proses bisnis, peneliti mengumpulkan data dengan wawancara.

4.1.1.1 Proses Bisnis *As-Is* Penerimaan Pasien Baru

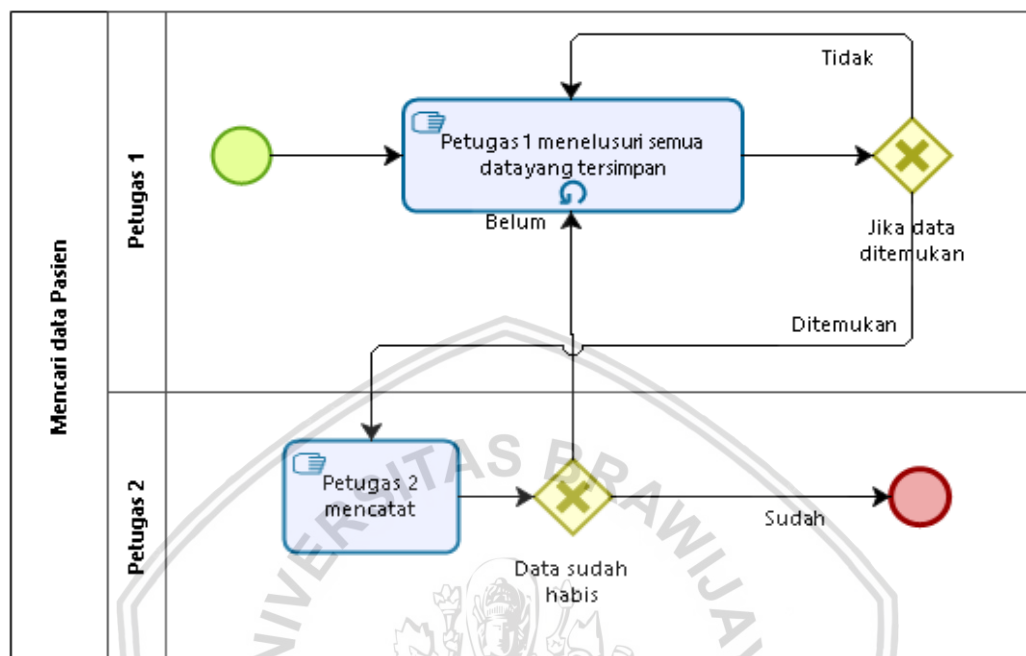
Proses bisnis dimulai dengan calon pasien yang datang, dan petugas mengarahkan calon pasien ke konselor. Lalu konselor mewawancarai dan memberikan beberapa *form* untuk diisi lalu setelah itu *assesor* mengambil urine dan (atau) darah pasien untuk diuji. Sementara menunggu hasil pengujian konselor melakukan *asessement* lanjutan dengan pertanyaan yang lebih spesifik lalu setelah selesai. Konselor berdiskusi untuk jadwal konseling berikutnya, pasien menerima hasil diskusi dan berakhir. Pemodelan proses bisnis dapat dilihat pada Gambar 4.1



Gambar 4.1 Proses Bisnis As-Is Penerimaan Pasien Baru

4.1.1.2 Proses Bisnis As-Is Mencari Data Pasien

Proses bisnis dimulai dengan petugas mencari data yang dicari secara manual dengan bantuan petugas lainnya, jika data ditemukan maka petugas lainnya mencatat hingga semua data ditelusuri.

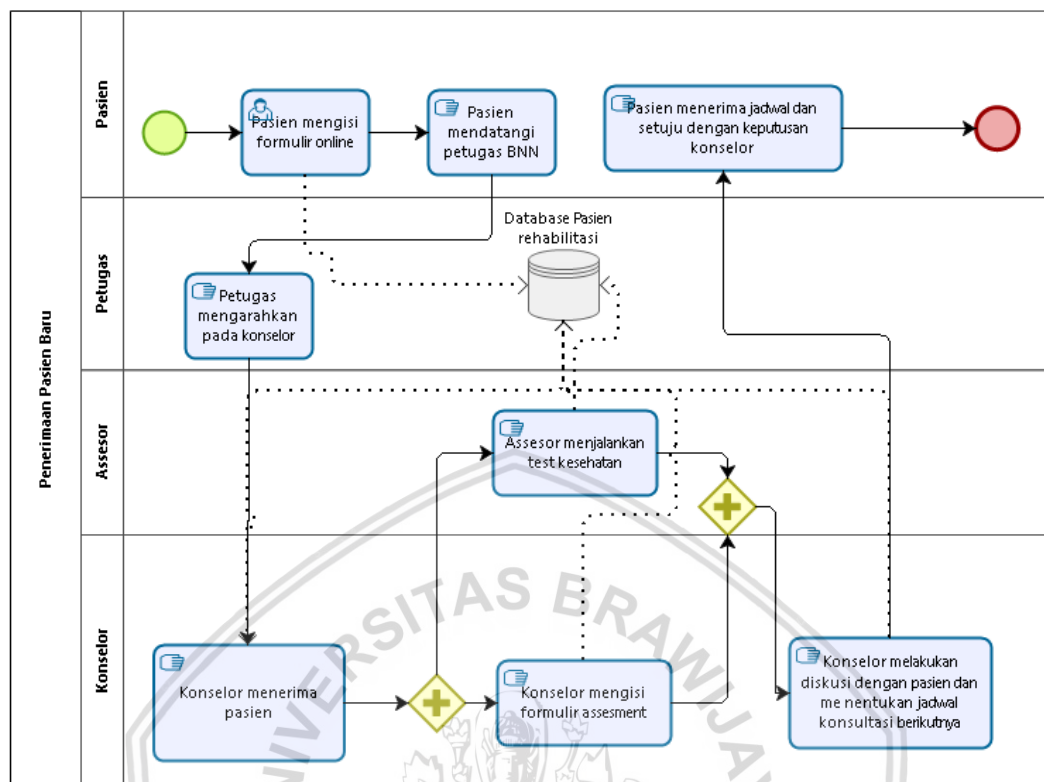


Gambar 4.2 Proses Bisnis As-Is Mencari Data Pasien

4.1.2 Proses Bisnis To-Be

Proses bisnis di bawah ini merupakan usulan yang ditawarkan kepada pemangku kepentingan (*to-be*), berdasarkan hasil analisis dari penulis yaitu menentukan aktivitas yang membutuhkan sistem informasi dan mengeliminasi aktivitas yang redundan dan tidak efektif. Proses bisnis To-Be dimodelkan untuk menunjukkan gambaran perubahan yang diusulkan pada penelitian ini yang digunakan sebagai masukan aktivitas selanjutnya.

4.1.2.1 Proses Bisnis To-Be Penerimaan Pasien Baru



Gambar 4.3 Proses Bisnis To-Be Penerimaan Pasien Baru

Pasien mengisi data melalui mengisi *form* yang ada di situs web secara online, lalu ketika sampai di BNN pasien melakukan verifikasi kartu identitas apakah sama yang seperti diunggah di *website*. Lalu selanjutnya petugas mengarahkan pada konselor, dan pengumpulan data lainnya seperti wawancara dilakukan sementara *asesor* mengecek kesehatan pasien (melalui urine atau darah), dilanjutkan dengan diskusi bersama konselor untuk menentukan jadwal berikutnya untuk konseling, dan terakhir pasien menerima keputusan dari konselor.

Tabel 4.1 Perbandingan waktu proses bisnis *to-be* dan *as-is*

<i>Proses Bisnis</i>	<i>Instances completed</i>	<i>Instances started</i>	<i>Min. time (s)</i>	<i>Max. time (s)</i>	<i>Avg. time (s)</i>	<i>Total time (s)</i>
<i>to-be</i>	3	3	5470	5470	5470	18210
<i>as-is</i>	3	3	6960	6960	6960	23580

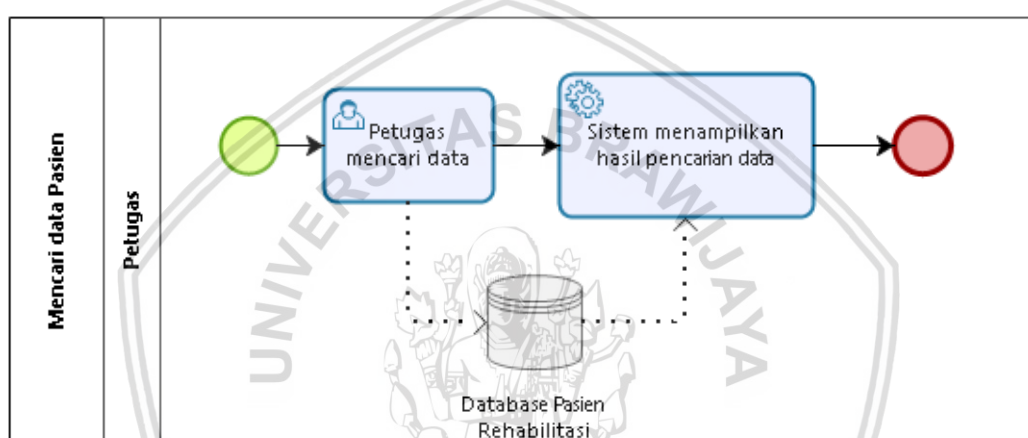
Tabel 4.2 Perubahan aktivitas Proses Bisnis Penerimaan Klien Baru

Unit Bisnis	Proses Bisnis As-Is	Proses Bisnis To-Be	Keterangan
Konselor	Konselor menerima pasien melakukan	Konselor menerima pasien	Diubah

Tabel 4.2 Perubahan aktivitas Proses Bisnis Penerimaan Klien Baru (Lanjutan)

	wawancara, memberikan formulir pada pasien untuk diisi		
Konselor	Konselor menerima formulir yang terisi	-	Dieliminasi
Pasien	Pasien mengisi formulir	Pasien mengisi formulir online	Diubah dan urutan aktivitas didahulukan

4.1.2.2 Proses Bisnis To-Be Mencari Data Pasien



Gambar 4.4 Proses Bisnis To-Be Mencari data pasien

Proses bisnis dimulai dengan petugas mencari data melalui sistem dan sistem menampilkan hasil pencarian, karena data disimpan di *database* maka pencarian menjadi lebih cepat dan efektif.

4.1.2.3 Analisis Waktu Proses Bisnis To-Be Mencari data pasien

Berikut hasil simulasi analisis waktu dari proses bisnis usulan penerimaan pasien baru, proses bisnis ini merupakan simulasi jika produk (sistem) diterapkan oleh organisasi. Tabel 4.3 menunjukkan perbandingan analisis waktu jika sistem diterapkan pada organisasi, bagaimana pengurangan waktu yang signifikan dari 4860 detik menjadi hanya 21 detik.

Tabel 4.3 Perbandingan waktu proses bisnis to-be dan as-is

<i>Proses Bisnis</i>	<i>Instances completed</i>	<i>Instances started</i>	<i>Min. time (s)</i>	<i>Max. time (s)</i>	<i>Avg. time (s)</i>	<i>Total time (s)</i>
<i>to-be</i>	3	3	7	7	7	21
<i>as-is</i>	3	3	360	4140	1620	4860

Tabel 4.4 Perubahan aktivitas Proses Bisnis Cari Data

Unit Bisnis	Proses Bisnis As-Is	Proses Bisnis To-Be	Keterangan
Petugas 1	Petugas 1 menelusuri semua data yang tersimpan	Petugas mencari data	Diubah
Petugas 2	Petugas 2 mencatat	-	Dieliminasi

4.2 Analisis Persyaratan

4.2.1 Identifikasi Pemangku Kepentingan

Untuk mengetahui hubungan pemangku kepentingan dengan sistem maka Identifikasi tipe pemangku kepentingan dilakukan, informasi ini didapatkan dari hasil analisis wawancara saat validasi masalah bersama beberapa staf rehabilitasi, hasil identifikasi adalah pengelompokan pemangku kepentingan berdasarkan hubungan dengan sistem dan karakteristiknya yang berguna untuk kegiatan analisis permasalahan.

Tabel 4.5 Tipe Pemangku Kepentingan

Tipe Pemangku Kepentingan	Deskripsi	Pemangku Kepentingan
Pengguna	Individu yang mempunyai interaksi dengan sistem, aktor dalam <i>use case</i> (yang akan menggunakan sistem)	Pasien Konselor Petugas
Pengembang	Individu yang mengembangkan sistem informasi data pasien	Peneliti
Pihak yang berwenang	Organisasi atau individu yang memberikan dukungan dalam memberikan informasi guna pengembangan sistem informasi yang sesuai dengan regulasi dan aturan yang berlaku	Seksi Rehabilitasi BNN Kota Malang
Pelanggan	Organisasi atau individu yang merasakan hasil dari sistem informasi pasien	BNN Kota Malang

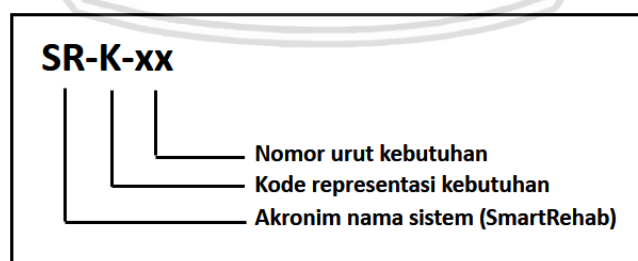
4.2.2 Analisis Masalah

Analisis masalah dilakukan untuk mengerti dan lebih memahami masalah yang terjadi di organisasi, analisis masalah didapatkan dari hasil analisis wawancara staf rehabilitasi dan staf bagian umum, *problem statement* ini akan di analisis dan menjadi identifikasi kebutuhan. Penjelasan *problem statement* terdapat pada tabel berikut.

Tabel 4.6 Problem Statement

The Problem Of	<ul style="list-style-type: none"> • Pencarian data pasien rehabilitasi masih secara manual, sehingga banyak waktu yang digunakan • Data pasien yang masih <i>hardcopy</i>, sulit untuk dirawat dan berbagai ancaman lain jika data hanya disimpan secara fisik • Pengulangan data dalam pengisian formulir oleh pasien
Affects	Pengguna, Pihak yang berwenang, Pelanggan
The Impact of which is	<ul style="list-style-type: none"> • Sumber daya banyak terkuras untuk mencari data • Jika terjadi penambahan data baru, maka harus di <i>update</i> secara manual, sehingga data tidak terbaru • Data yang hilang atau kurang menyebabkan informasi yang kurang valid dan mengurangi integritas BNN dalam mengelola data pasien • Proses Rehabilitasi terganggu, dan data yang akan dilaporkan berkurang kualitasnya
A succesful solution would	Sistem informasi untuk mengelola data pasien rehabilitasi, untuk memudahkan pencarian data pasien rehabilitasi

4.2.3 Identifikasi Kebutuhan Pengguna



Gambar 4.5 Keterangan Kodifikasi Kebutuhan Pengguna

Identifikasi kebutuhan pengguna diperoleh dari hasil analisis *problem statement*, Hasil identifikasi yang merupakan kebutuhan digunakan untuk kegiatan analisis fitur untuk solusi bagi permasalahan. Setiap kebutuhan pengguna pada tabel di bawah ini diberi kode sebagai identitas, keterangan kodifikasi dapat dilihat pada gambar berikut.

Tabel 4.7 Hasil Identifikasi Kebutuhan Pengguna

Kode Kebutuhan	Kebutuhan Pengguna	Pemangku kepentingan	Situasi Saat Ini	Solusi yang ditawarkan
SR-K-01	Sistem bisa menyimpan data pasien	Petugas, Konselor	Data disimpan secara fisik (<i>hardcopy</i>) di dalam lemari	Sistem informasi untuk menyimpan data di dalam <i>database</i>
SR-K-02	Sistem bisa mencetak <i>form</i> yang sudah terisi	Pasien, Petugas dan Konselor	<i>Form</i> kosong di cetak lalu di isi secara manual	Sistem Informasi yang dapat mencetak <i>form</i> yang <i>field-field</i> nya telah terisi dengan data yang dimasukkan oleh pasien
SR-K-03	Sistem yang bisa membuat pasien tidak menuliskan informasi yang sama berulang-ulang	Pasien	<i>Form</i> di isi secara manual sehingga data yang berulang tetap ditulis kembali	Sistem informasi yang dapat menyimpan data masukan, sehingga dengan sekali <i>input</i> , <i>field-field</i> yang sama juga terisi sehingga tidak perlu memasukkannya lagi
SR-K-04	Sistem yang bisa mengelola data pasien	Petugas, Konselor	Data pasien dikelola, dipelihara secara manual sehingga rentan terhadap hilangnya data, kurang lengkapnya data	Sistem informasi untuk menambah, mengubah hingga menghapus data tanpa kontak langsung dokumen fisik
SR-K-05	Sistem yang pencarian datanya mudah dan cepat	Petugas, Konselor	Pencarian dilakukan secara manual menggunakan ketelitian manusia,	Sistem informasi yang dapat melakukan pencarian data yang tersimpan

Tabel 4.7 Hasil Identifikasi Pengguna (Lanjutan)

			aktivitas ini memakan banyak waktu dan menimbulkan kejenuhan, data yang dicari bisa saja terlewat atau akurasi kurang	secara cepat dan akurat
SR-K-06	Sistem yang mudah digunakan dan mudah dipelajari	Petugas, Konselor dan Pasien	Pencatatan data pasien masih dengan cara konvensional	Sistem Informasi yang memudahkan penggunaanya dalam menyelesaikan aktivitas
SR-K-07	Sistem bisa diakses dari komputer mana saja, termasuk platform <i>mobile</i>	Pasien, Petugas dan Konselor	-	<i>Frontend</i> sistem dirancang dengan pendekatan <i>grid</i> RWD (<i>Responsive Web Design</i>)

4.2.4 Identifikasi Pengguna

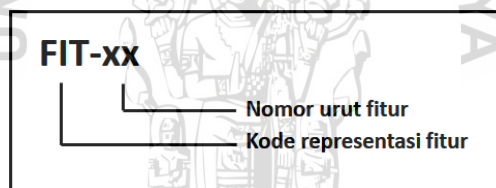
Identifikasi pengguna merupakan tahap untuk mengetahui pengguna yang akan menggunakan dan berinteraksi langsung dengan sistem, dimana ada 2 tipe pemangku kepentingan yaitu pengguna dan pengembang, pengguna terdiri 3 tipe yaitu pasien, petugas, dan konselor yang telah di uraikan dalam Tabel 4.8. Hasil identifikasi akan digunakan untuk tahap pembuatan *use case*.

Tabel 4.8 Hasil Identifikasi Pengguna

Tipe Pemangku Kepentingan	Tipe Pengguna	Deskripsi
Pengguna	Pasien	Pasien dari berbagai latar belakang, pengguna yang ingin sembuh, bisa jadi rujukan dari rumah sakit atau tertangkap tangan dari operasi polisi, data dari pasien yang disimpan di dalam <i>database</i> sistem.

Tabel 4.8 Hasil Identifikasi Pengguna (Lanjutan)

	Petugas	Petugas mengantarai antara pasien dan staf yang lain (konselor), Mengambil informasi dari pasien terkait latar belakang kesehatan, memiliki kewenangan untuk mencari data. Bertugas mengelola data pasien.
	Konselor	Konselor mempunyai kemampuan untuk menyembuhkan secara psikologis dan memiliki pengetahuan klinis, menggunakan kertas dan alat tulis sebagai media hasil konseling dari pasien, tatap muka menjadi kebutuhan dari konselor untuk menggali data dari pasien, mengelola data pasien yang terkait dengan hasil konseling dan data pribadi lainnya.
Pengembang	Peneliti	Peneliti memiliki akses beberapa data yang diperbolehkan untuk menguji coba beberapa modul dalam sistem demi pengembangan sistem informasi data pasien.

4.2.5 Identifikasi Fitur**Gambar 4.6 Kodifikasi fitur**

Tabel di bawah ini berisi solusi yang ditunjukkan dengan deskripsi singkat yang diidentifikasi menjadi sebuah fitur. Gambar 4.6 merupakan kodifikasi fitur. Fitur-fitur akan digunakan untuk tahap menggali syarat fungsional dan non-fungsional sistem, deskripsi dapat dilihat pada Tabel 4.9.

Tabel 4.9 Hasil Identifikasi Fitur

Kode Fitur	Deskripsi
FIT-01	Sistem dapat mengetahui pengguna sistem yang memiliki kewenangan yang berbeda, beberapa data pribadi hanya bisa dikelola oleh konselor.
FIT-02	Sistem dapat mengelola data pasien, menambah, mencari, melihat dan menghapus data sesuai dengan regulasi dan aturan yang berlaku.
FIT-03	Sistem menyediakan <i>form</i> bagi pasien untuk memasukkan data yang dibutuhkan.

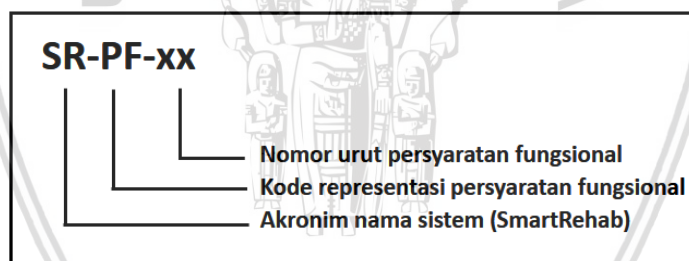
Tabel 4.9 Hasil Identifikasi Fitur (Lanjutan)

FIT-04	Sistem menyediakan layanan bagi data pasien yang tersimpan, agar bisa dicetak seperti formulir-formulir yang berlaku sesuai aturan
FIT-05	Sistem bisa diakses melalui beberapa <i>browser</i> dan mampu diakses menggunakan <i>platform mobile</i> dengan tampilan yang sesuai juga bagi tampilan <i>mobile</i> . Antarmuka sistem mudah dipelajari dan mudah digunakan.

Tabel 4.10 Hubungan Kebutuhan Pengguna dan Fitur

Kebutuhan Pengguna	Fitur
SR-K-06	FIT-01
SR-K-01	FIT-02
SR-K-04	
SR-K-05	
SR-K-03	FIT-03
SR-K-02	FIT-04
SR-K-07	FIT-05

4.2.6 Persyaratan Fungsional

**Gambar 4.7 Kodifikasi Persyaratan Fungsional**

Tahap ini menjelaskan bagaimana relasi persyaratan fungsional yang akan dikembangkan dan fitur yang ditawarkan sistem nantinya dan bagaimana kesesuaiannya dengan fitur yang memenuhi persyaratan fungsional yang merupakan kebutuhan pengguna, dapat dilihat di Tabel 4.11.

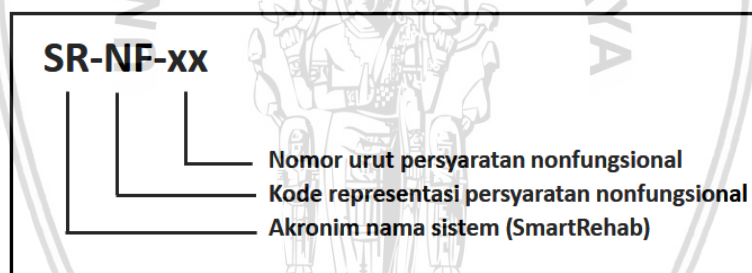
Tabel 4.11 Persyaratan Fungsional

Kode Fitur	Kode Persyaratan Fungsional	Deskripsi
FIT-01	SR-PF-01	Pengguna dapat masuk ke dalam sistem dengan menggunakan autentikasi <i>username</i> dan <i>password</i> yang tersimpan dalam <i>database</i> sistem
		Pengguna bisa mengganti <i>password</i>

Tabel 4.11 Persyaratan Fungsional (Lanjutan)

	SR-PF-02	Pengguna bisa keluar dari sistem
FIT-02	SR-PF-03	Sistem dapat mengubah data pasien sesuai prosedur yang berlaku
	SR-PF-04	Sistem menyimpan perubahan data dalam <i>database</i>
	SR-PF-05	Sistem dapat menampilkan hasil pencarian sesuai dengan data yang ingin dicari
FIT-03	SR-PF-06	Sistem menyediakan <i>form</i> yang dapat menambahkan data pasien sesuai dengan prosedur yang berlaku
	SR-PF-07	Sistem dapat meng-validasi apakah data yang dimasukkan sesuai dengan aturan yang berlaku
	SR-PF-08	Sistem menyimpan penambahan data dalam <i>database</i>
FIT-04	SR-PF-09	Sistem mengambil informasi yang dibutuhkan dari <i>database</i>
	SR-PF-10	Sistem menaruh data dalam dokumen elektronik lalu mencetaknya

4.2.7 Persyaratan Non-fungsional

**Gambar 4.8 Kodifikasi Persyaratan Non-fungsional**

Persyaratan nonfungsional merupakan persyaratan sistem untuk mendukung berjalannya sistem nantinya tetapi bukan merupakan persyaratan fungsional, persyaratan non-fungsional didapatkan dari hasil wawancara dengan staf rehabilitasi, kodifikasi persyaratan non-fungsional dapat dilihat pada Gambar 4.8 sebagai identitas, dan dikelompokkan sesuai fitur yang berkaitan. Untuk menguji apakah persyaratan nonfungsional berhasil diterapkan maka nantinya akan dilakukan pengujian sesuai persyaratan nonfungsional yang dirancang.

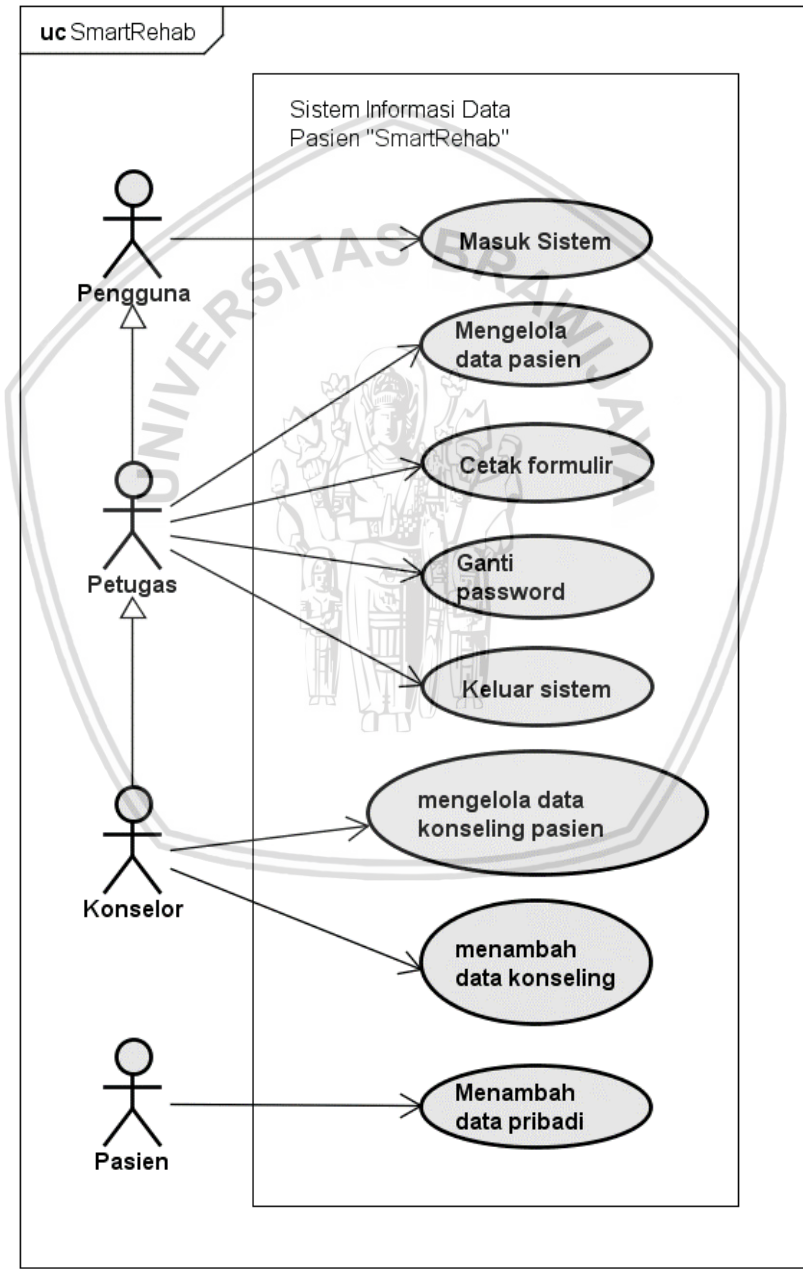
Tabel 4.12 Persyaratan Non-fungsional

Kode Fitur	Kode Persyaratan Non-fungsional	Deskripsi
FIT-05	SR-NF-01	Sistem dapat memudahkan penggunaanya dalam menyelesaikan aktivitas, dan sistem mempunyai antarmuka pengguna yang mudah dipelajari.

Tabel 4.12 Persyaratan Non-fungsional (Lanjutan)

	SR-NF-02	Sistem bisa diakses melalui beberapa browser dan mampu diakses menggunakan platform <i>mobile</i> dengan tampilan yang sesuai juga bagi tampilan <i>mobile</i> (tidak ada yang hilang/terpotong/tidak bisa diklik dsb.).
--	----------	--

4.3 Pemodelan Use Case



Gambar 4.9 Use-Case Diagram Sistem Informasi “SmartRehab”

Use case diagram berguna untuk mengenali aktor dan *use case* yang merupakan kelakuan atau *behaviour* dari sistem informasi, dan spesifikasi masing-masing *use case* yang berisi skenario cerita bagaimana aktor menyelesaikan suatu objektif. Identifikasi aktor dikelompokkan berdasarkan karakteristiknya, tahap ini dilakukan setelah menyelesaikan tahap analisis persyaratan. Penentuan aktor berdasarkan pada Identifikasi pengguna dalam tahap analisis persyaratan, masing-masing *use case* yang telah teridentifikasi diberikan penjelasan singkat, urutan tahap untuk menyelesaikan *use case* serta kondisi sesudah dan sebelum yang akan didokumentasikan dalam spesifikasi *use-case*, spesifikasi *use case* merupakan syarat bagi pemodelan aktivitas ke dalam *activity* diagram.

4.3.1 Use Case Diagram

Use case yang berhasil dikembangkan/ditemukan dikomparasikan dengan solusi yang ditawarkan dalam aktivitas - aktivitas proses bisnis *to-be*, untuk menunjukkan apakah solusi yang ditawarkan dari *Use Case* berkaitan dengan proses bisnis *to-be*.

Tabel 4.13 Hubungan aktivitas Proses Bisnis *to-be* dengan *Use Case*

Proses Bisnis	Aktivitas	<i>Use Case</i>
Penerimaan Pasien baru	Pasien mengisi formulir online	Menambah data pribadi
	Konselor mengisi formulir <i>assesment</i>	Mengelola data pasien
	Konselor menerima pasien	Menambah data konseling pasien
		Cetak Formulir
Mencari data pasien	Petugas mencari data	Mengelola data pasien

Kemudian menghubungkan hubungan antara *use case* dan tipe pemangku kepentingan untuk memahami apa saja yang pengguna dapat lakukan terhadap sistem informasi.

Tabel 4.14 Hubungan *Use Case* dan Pemangku Kepentingan

<i>Use Case</i>	Pengguna	Tipe pemangku kepentingan
Masuk Sistem	Konselor, Petugas	Pengguna
Mengelola data pasien	Konselor, Petugas	
Mengelola data konseling pasien	Konselor	
Cetak formulir	Konselor, Petugas	
Ganti <i>password</i>	Konselor, Petugas	

Tabel 4.14 Hubungan Use Case dan Pemangku Kepentingan (Lanjutan)

Menambah data konseling	Konselor, Petugas	
Keluar sistem	Konselor, Petugas	
Menambah data pribadi	Pasien	

Setelah itu *use case* dihubungkan dengan fitur hasil analisis persyaratan agar tampak bagaimana pemodelan memenuhi fitur yang ditawarkan oleh sistem.

Tabel 4.15 Hubungan Use Case dan Fitur

<i>Use Case</i>	Fitur
Masuk Sistem	FIT-01
Mengelola data pasien	FIT-02
Mengelola data konseling pasien	
Menambah data konseling	
Cetak formulir	FIT-04
Menambah data pribadi	FIT-03

4.3.2 Deskripsi Aktor

Deskripsi aktor berisi deskripsi singkat tentang bagaimana aktor-aktor yang telah teridentifikasi menjalankan sistem informasi, tanggung jawab beserta tujuan dari aktor dapat dilihat pada tabel berikut.

Tabel 4.16 Deskripsi Aktor

Nama Aktor	Deskripsi
Pasien	Pasien dari berbagai latar belakang, mantan pengguna atau orang yang sedang menjalani rehabilitasi rawat jalan.
Petugas	Petugas mengantarai antara pasien dan staf yang lain (konselor), Mengambil informasi dari pasien terkait latar belakang kesehatan.
Konselor	Konselor mempunyai kemampuan untuk menyembuhkan secara psikologis dan memiliki pengetahuan klinis, menggunakan kertas dan alat tulis sebagai media hasil konseling dari pasien, tatap muka menjadi kebutuhan dari konselor untuk menggali data dari pasien.

4.3.3 Spesifikasi Use Case

Bagian ini berisi cerita skenario setiap *use case* yang dibagi menjadi alur utama, alur alternatif dan kondisi sebelum dan sesudah *use case* dijalankan oleh aktor, hasil spesifikasi *use case* akan dimodelkan ke *activity diagram*.

4.3.3.1 Spesifikasi Use Case Masuk Sistem

Pada spesifikasi *use case* masuk sistem ini terdapat tujuan, kondisi sesudah dan sebelum pengguna menjalankan *use case* ini beserta tahap-tahap dan alur alternatif jika pengguna tidak dapat masuk sistem. Spesifikasi *use case* dapat dilihat Tabel 4.17 di bawah ini.

Tabel 4.17 Spesifikasi Use Case Masuk Sistem

Brief Description	Spesifikasi <i>use case</i> masuk sistem menjelaskan bagaimana pengguna teridentifikasi dan bisa masuk ke dalam sistem
Actor	Pengguna
Pre-Condition	<ul style="list-style-type: none"> Halaman <i>Login</i> telah ditampilkan oleh sistem
Post-Condition	<i>Login</i> status sukses dan terotentikasi sebagai petugas/konselor
Basic Flow	<ol style="list-style-type: none"> Aktor memasukkan <i>username</i> dan <i>password</i> ke dalam <i>form</i> Aktor mengirim data <i>username</i> dan <i>password</i> Sistem mengenali pengguna {mengenali pengguna} Lalu sistem memasukkan pengguna sesuai tipe penggunanya
Alternative Flow	<p>A1. Menangani kegagalan mengenali pengguna</p> <p>Pada {mengenali pengguna}, jika mengalami kegagalan maka sistem menampilkan pesan kesalahan (kesalahan <i>input</i>, <i>username</i> tidak tersedia dan <i>password</i> salah atau kesalahan sistem), lalu <i>use case</i> selesai.</p>

4.3.3.2 Spesifikasi Use Case Mengelola Data Pasien

Pada spesifikasi *use case* (Tabel 4.18) menunjukkan tujuan dan siapa aktor dari *use case* ini beserta kondisi yang harus dipenuhi agar bisa menjalankan *use case* ini, beserta alur alternatif (kemungkinan alur lain).

Tabel 4.18 Spesifikasi Use Case Mengelola Data Pasien

Brief Description	Spesifikasi <i>use case</i> ini menjelaskan cerita atau alur aktor dalam menjalankan <i>use case</i> Mengelola data pasien hingga objektif tercapai, beserta alternatif <i>flow</i>
Actor	Petugas, Konselor
Pre-Condition	<i>Login</i> status sukses dan terotentikasi sebagai petugas/konselor

Tabel 4.18 Spesifikasi Use Case Mengelola Data Pasien (Lanjutan)

Post-Condition	<ul style="list-style-type: none"> • Sistem menyimpan perubahan data pasien • Sistem menampilkan perubahan data pasien
Basic Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu mengelola pasien 2. Sistem menampilkan data pasien 3. Aktor melakukan pencarian data pasien <ol style="list-style-type: none"> a. Memasukkan nama atau ciri-ciri lainnya pada kolom pencarian 4. Sistem menampilkan hasil pencarian 5. Aktor memilih pasien mana yang ingin diubah datanya 6. Aktor melakukan perubahan data 7. Aktor mengirim data 8. Sistem menyimpan perubahan data 9. Sistem menampilkan data pasien yang sudah diubah
Alternative Flow	<p>A1. Menangani data pasien yang tidak muncul Pada saat sistem menampilkan data, sistem memberi pesan jika data tidak tersedia.</p> <p>A2. Menangani aktor yang batal melakukan perubahan data Perubahan yang dilakukan tidak disimpan dan data tidak diubah, dan kembali pada tampilan menampilkan data pasien.</p> <p>A3. Menangani kegagalan pada saat menyimpan data Sistem akan menampilkan pesan kesalahan (kesalahan <i>input</i>, kesalahan sistem, dsb.).</p>

4.3.3.3 Spesifikasi Use Case Mengelola Data Konseling Pasien

Pada Tabel 4.19 menunjukkan bagaimana konselor menjalankan *use case* mengelola data konseling pasien, tujuan dan kondisi yang harus dipenuhi beserta alur alternatif untuk kasus di luar *basic flow*.

Tabel 4.19 Spesifikasi Use Case Mengelola Data Konseling Pasien

Brief Description	Menjelaskan bagaimana aktor mengelola data konseling pasien (mengubah dan menghapus), beserta <i>alternative flow</i> -nya
--------------------------	--

Tabel 4.19 Spesifikasi *Use Case* Mengelola Data Konseling Pasien (Lanjutan)

Actor	Konselor
Pre-Condition	<i>Login</i> status sukses dan terotentikasi sebagai konselor
Post-Condition	<ul style="list-style-type: none"> • Perubahan data konseling tersimpan • Sistem menampilkan perubahan data konseling pasien
Basic Flow	<ol style="list-style-type: none"> 1. <i>Use case</i> dipicu oleh Aktor yang memilih menu mengelola data konseling pasien 2. Sistem menunjukkan data pasien pada aktor 3. Aktor melakukan pencarian data pasien <ol style="list-style-type: none"> a. Memasukkan nama atau ciri-ciri lainnya pada kolom pencarian 4. Sistem menampilkan hasil pencarian 5. Konselor memilih pasien yang akan dikelola data konselingnya 6. Konselor memilih data konseling yang akan diubah 7. Konselor melakukan perubahan data 8. Sistem menyimpan data konseling 9. Sistem menampilkan pesan data berhasil diubah
Alternative Flow	<p>A1. Menangani data pasien yang tidak muncul Pada saat sistem menampilkan data, sistem memberi pesan jika data tidak tersedia</p> <p>A2. Menangani konselor yang batal melakukan perubahan data Perubahan yang dilakukan tidak disimpan dan data tidak diubah, dan kembali pada tampilan menampilkan data pasien.</p> <p>A3. Menangani kegagalan pada saat menyimpan data Sistem akan menampilkan pesan kesalahan (kesalahan <i>input</i>, kesalahan sistem, dsb.).</p>

4.3.3.4 Spesifikasi Use Case Cetak Formulir

Spesifikasi *use case* berikut menjelaskan alur cerita bagaimana aktor mencetak formulir hingga kondisi yang harus dipenuhi agar *use case* dapat dijalankan, untuk lengkapnya dapat dilihat pada Tabel 4.20.

Tabel 4.20 Spesifikasi Use Case Cetak Formulir

Brief Description	Spesifikasi <i>use case</i> ini menjabarkan cerita bagaimana para aktor menyelesaikan tahap-tahap dalam mencetak formulir beserta alternatif <i>flownya</i>
Actor	Konselor, Petugas
Pre-Condition	<i>Login</i> status sukses dan terotentikasi sebagai petugas/konselor
Post-Condition	Formulir yang terisi informasi pasien telah selesai dicetak
Basic Flow	<ol style="list-style-type: none"> 1. <i>Use case</i> dipicu oleh Aktor yang memilih menu Cetak formulir pasien 2. Sistem menampilkan data pasien 3. Aktor memilih pasien yang akan dicetak formulirnya 4. Sistem menampilkan formulir dari pasien yang dipilih 5. Aktor memilih formulir apa yang akan dicetak 6. Sistem mencetak formulir yang dipilih aktor
Alternative Flow	<p>A1. Menangani data pasien yang tidak muncul</p> <p>Pada saat sistem menampilkan data, sistem memberi pesan jika data tidak tersedia, lalu <i>use case</i> selesai</p> <p>A2. Menangani formulir yang tidak tercetak</p> <p>Sistem akan memberikan pesan bahwa ada kesalahan pada sistem atau kesalahan teknis, pesannya adalah dimohon aktor mengecek keadaan <i>printer</i> apakah sudah tersambung dengan baik (keadaan kertas, tinta, dsb), lalu <i>use case</i> selesai</p> <p>A3. Menangani formulir yang beberapa informasinya tidak tersedia</p> <ol style="list-style-type: none"> 1. Sistem akan memberitahukan bahwa beberapa informasi tidak tersedia (belum dimasukkan) pada formulir

Tabel 4.20 Spesifikasi *Use Case* Cetak Formulir (Lanjutan)

	<ol style="list-style-type: none"> Aktor memilih apakah tetap di cetak atau batal dicetak Use case selesai <p>A4. Menangani aktor yang batal mencetak formulir</p> <p>Aktor dikembalikan ke tahap melihat data pasien, dan formulir batal dicetak, lalu <i>use case</i> selesai</p>
--	--

4.3.3.5 Spesifikasi *Use Case* Ganti Password

Pada Tabel 4.21 menjelaskan spesifikasi *use case* Ganti password, untuk mengetahui bagaimana tahap untuk menyelesaikan, kondisi yang harus dipenuhi, tujuan dan alur alternatif dapat dilihat pada tabel.

Tabel 4.21 Spesifikasi *Use Case* Ganti Password

Brief Description	Pada spesifikasi <i>use case</i> ini akan menampilkan tahap bagaimana para aktor mengganti <i>password</i> beserta alternatif <i>flownya</i>
Actor	Konselor, Petugas
Pre-Condition	<i>Login</i> status sukses dan terotentikasi sebagai petugas/konselor
Post-Condition	<i>Password</i> sudah berubah, aktor tidak bisa masuk sistem dengan <i>password</i> lama, melainkan <i>password</i> yang baru
Basic Flow	<ol style="list-style-type: none"> Aktor memilih menu ganti password Sistem menampilkan <i>form</i> untuk memasukkan <i>password</i> yang lama dan <i>password</i> yang baru Aktor mengisi <i>form</i> lalu mengirimnya Sistem mengautentikasi <i>password</i> lama Sistem mengganti <i>password</i> Sistem menampilkan pesan bahwa <i>password</i> telah sukses diganti
Alternative Flow	<p>A1. Menangani <i>password</i> lama yang salah</p> <p>Sistem akan menampilkan pesan bahwa <i>password</i> lama salah, <i>password</i> tidak terganti, lalu <i>use case</i> selesai</p> <p>A2. Menangani <i>password</i> baru yang tidak cocok dengan konfirmasi <i>password</i></p>

Tabel 4.21 Spesifikasi Use Case Ganti Password (Lanjutan)

	<p>Sebuah pesan akan ditampilkan oleh sistem: <i>password</i> dan konfirmasi <i>password</i> tidak cocok, <i>password</i> tidak terganti, lalu <i>use case</i> selesai.</p> <p>A3. Menangani <i>password</i> baru yang sama dengan <i>password</i> lama</p> <p>Sistem akan menampilkan pesan bahwa <i>password</i> baru tidak boleh sama dengan <i>password</i> lama, lalu <i>use case</i> selesai</p>
--	---

4.3.3.6 Spesifikasi Use Case Menambah Data Konseling

Pada spesifikasi *use case* ini menjelaskan bagaimana konselor menambah data konseling, penjelasan kondisi yang harus dipenuhi dan tujuan dari *use case* ini dapat melihat pada Tabel 4.22.

Tabel 4.22 Spesifikasi Use Case Menambah Data Konseling

Brief Description	Pada spesifikasi <i>use case</i> ini akan menyajikan alur bagaimana konselor menambah data konseling pasien
Actor	Konselor
Pre-Condition	<i>Login</i> status sukses dan terotentikasi sebagai konselor
Post-Condition	<ul style="list-style-type: none"> • Perubahan data konseling pasien (penambahan, perubahan, penghapusan) tersimpan • Sistem menampilkan perubahan data konseling pasien
Basic Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu menambah data konseling 2. Sistem menunjukkan data pasien 3. Konselor memilih pasien yang akan ditambah data konseling 4. Sistem menampilkan <i>form</i> untuk mengisi data konseling 5. Konselor mengisi data konseling 6. Konselor mengirim data 7. Sistem menyimpan data konseling 8. Sistem menampilkan pesan data berhasil ditambahkan
Alternative Flow	<p>A1. Menangani data pasien yang tidak muncul</p> <p>Pada saat sistem menampilkan data, sistem memberi pesan jika data tidak tersedia</p>

Tabel 4.22 Spesifikasi *Use Case* Menambah Data Konseling (Lanjutan)

	<p>A2. Menangani konselor yang batal melakukan penambahan data</p> <p>Penambahan yang dilakukan tidak disimpan dan data tidak ditambahkan, dan kembali pada tampilan menampilkan data pasien</p> <p>A3. Menangani kegagalan pada saat menyimpan data</p> <p>Sistem akan menampilkan pesan kesalahan (kesalahan <i>input</i>, kesalahan sistem, dsb.)</p>
--	--

4.3.3.7 Spesifikasi *Use Case* Menambah Data Pribadi

Tabel 4.23 menyajikan bagaimana aktor pasien menyelesaikan *use case* menambah data pribadi, alur-alur dan kondisi yang harus dipenuhi beserta tujuan alur alternatif dapat dilihat pada tabel.

Tabel 4.23 Spesifikasi *Use Case* Menambah Data Pribadi

Brief Description	Spesifikasi <i>use case</i> ini akan menampilkan bagaimana tahap-tahap dalam <i>use case</i> menambah data pribadi, apa yang dilakukan oleh pasien beserta alur alternatifnya.
Actor	Pasien
Pre-Condition	<ul style="list-style-type: none"> • Pasien terhubung dengan internet • Pasien terhubung dengan server sistem • Halaman pendaftaran telah ditampilkan sistem
Post-Condition	<ul style="list-style-type: none"> • Pasien menerima konfirmasi <i>email</i> • Pasien menerima sms/ panggilan dari staf rehabilitasi BNN Kota Malang
Basic Flow	<ol style="list-style-type: none"> 1. Pasien memilih menu daftar rehabilitasi 2. Sistem menampilkan <i>form</i> 3. Pasien mengisi semua <i>form</i> sesuai dengan petunjuk yang diberikan oleh sistem 4. Pasien mengirim data 5. Sistem menyimpan data pasien 6. Sistem mengirimkan konfirmasi email dan pemberitahuan pada staf rehabilitasi

Tabel 4.23 Spesifikasi *Use Case* Menambah Data Pribadi (Lanjutan)

Alternative Flow	<p>A1. Menangani kegagalan pada saat menyimpan data</p> <p>Sistem akan menampilkan pesan kesalahan (kesalahan <i>input</i>, kesalahan sistem, dsb.), lalu <i>use case</i> selesai</p> <p>A2. Menangani pasien yang batal mengisi formulir</p> <p>Sistem tidak akan menyimpan data dan mengosongkan <i>form</i>, lalu <i>use case</i> selesai</p> <p>A3. Menangani kegagalan mengirim konfirmasi email</p> <p>Sistem tetap menyimpan data pasien, dan secepatnya konfirmasi melalui sms akan dikirimkan oleh staf rehabilitasi</p>
-------------------------	--

4.3.3.8 Spesifikasi *Use Case* Keluar Sistem

Spesifikasi *use case* ini menjelaskan bagaimana para aktor keluar dari sistem, alur utama beserta alur alternatif dan kondisi yang harus dipenuhi untuk menyelesaikan *use case* ini ditunjukkan pada Tabel 4.24

Tabel 4.24 Spesifikasi *Use Case* Keluar Sistem

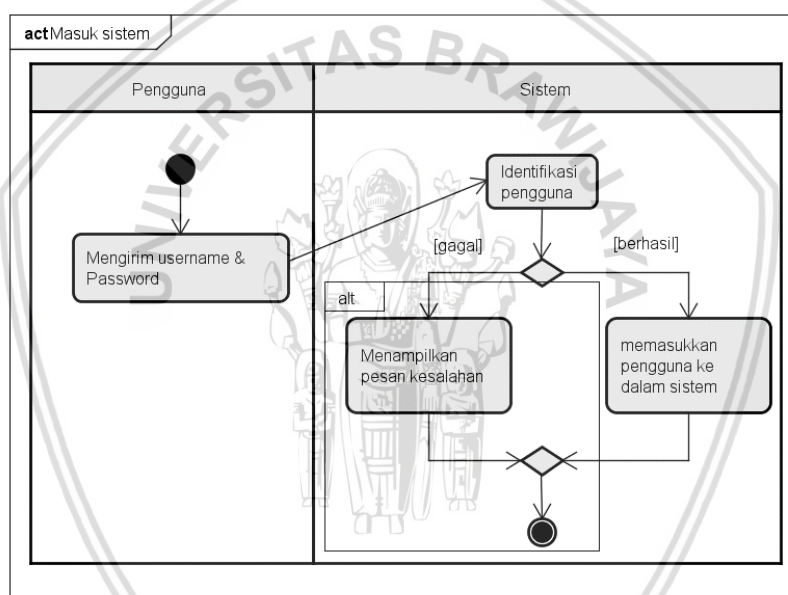
Brief Description	Pada spesifikasi <i>use case</i> ini menyajikan tahap-tahap bagaimana para aktor keluar dari sistem.
Actor	Konselor, Petugas
Pre-Condition	<i>Login</i> status sukses dan terotentikasi sebagai petugas/konselor
Post-Condition	<ul style="list-style-type: none"> <i>Login</i> status tidak aktif
Basic Flow	<ol style="list-style-type: none"> Aktor mengklik tombol <i>Logout</i> Sistem mengeluarkan aktor dari sistem Aktor kembali pada halaman <i>login</i>
Alternative Flow	<p>A1. Menangani aktor yang belum <i>logout</i> tapi sudah menutup <i>browser</i></p> <p><i>Cookie</i> akan kadaluwarsa dan aktor segera akan dikeluarkan dari sistem</p>

4.4 Pemodelan aktivitas

Pemodelan aktivitas dikembangkan berdasarkan alur yang tersedia pada spesifikasi *use case* (*Basic Flow*) dan *Alternative Flow* (jika tersedia) dan dimodelkan menggunakan *activity diagram*. Semua spesifikasi *use case* dimodelkan aktivitasnya, sehingga lebih mudah dipahami oleh pemangku kepentingan dan peneliti.

4.4.1 Pemodelan aktivitas Masuk Sistem

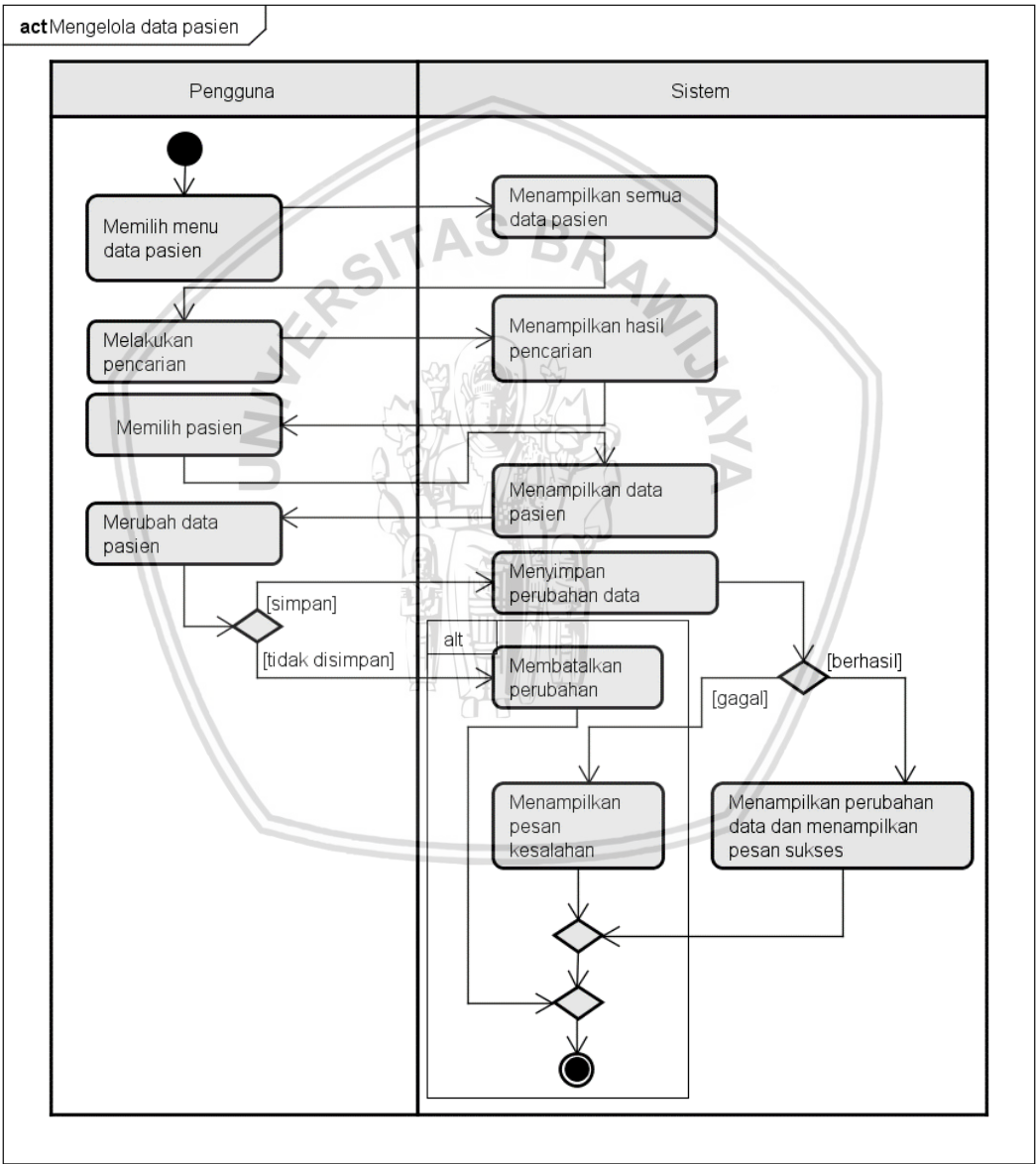
Pemodelan aktivitas masuk sistem pertama dimulai oleh pengguna yang mengirim *username* dan *password* ke sistem. Lalu sistem mengautentikasi apakah *username* tersebut tersedia dan *password*-nya benar, jika benar maka sistem akan mengizinkan masuk ke dalam sistem, dan jika tidak maka akan menampilkan pesan kesalahan (*Alternative Flow A1*), spesifikasi *use case* ditunjukkan pada tabel di atas Tabel 4.17 dan *Activity Diagram* ditunjukkan dengan di bawah (Gambar 4.10).



Gambar 4.10 Activity Diagram Masuk Sistem

4.4.2 Pemodelan aktivitas Mengelola Data Pasien

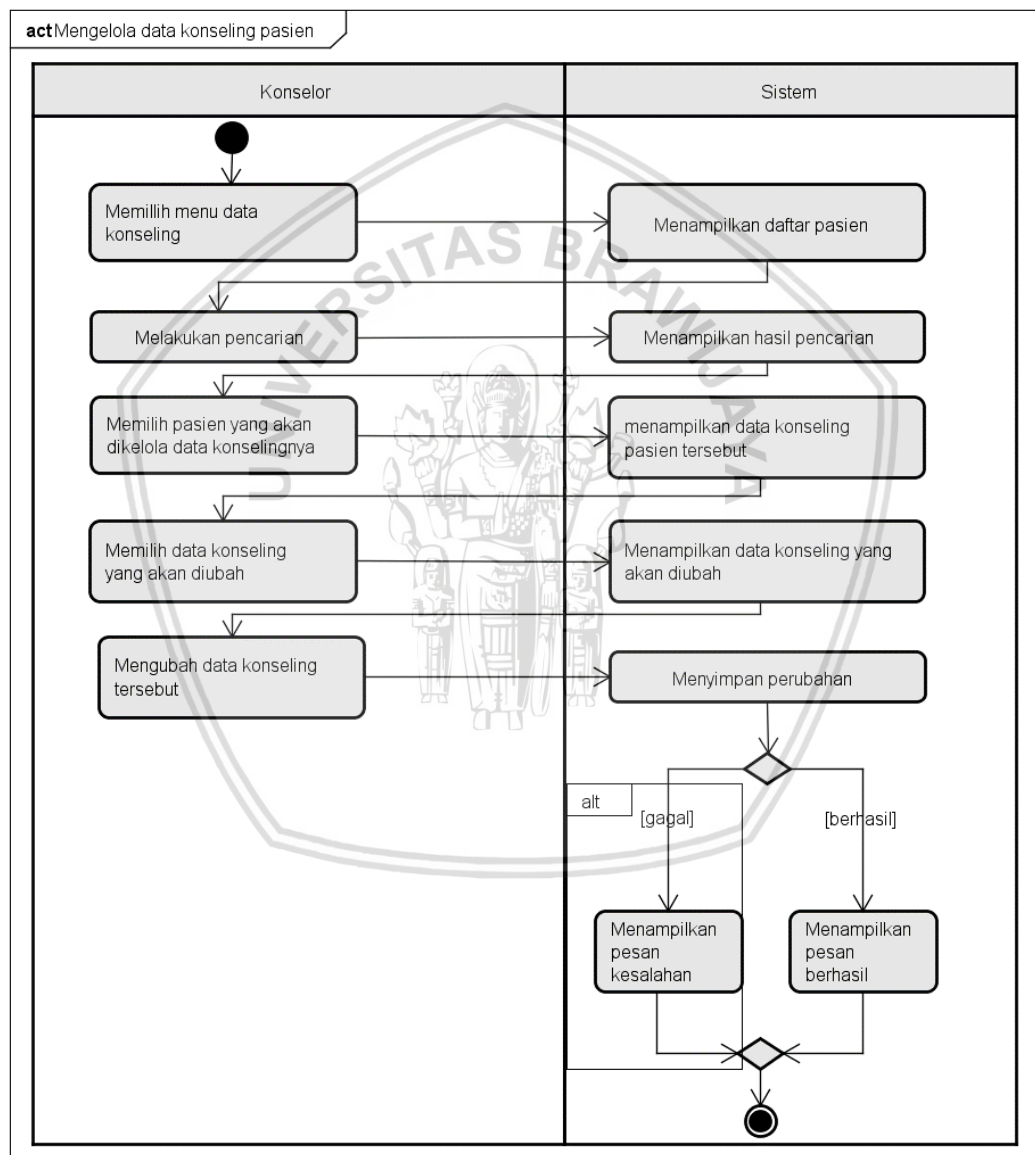
Pemodelan aktivitas mengelola data pasien dimulai oleh pengguna yang memilih mengelola data pasien, sistem menampilkan data pasien, selanjutnya pengguna memilih pasien mana yang akan diubah datanya, lalu sistem menampilkan *form* untuk mengubah data pasien, jika pengguna ingin membatalkan maka perubahan yang sudah dilakukan tidak diterapkan (*Alternative Flow A2*). Lalu jika gagal menyimpan data perubahan maka sistem akan menampilkan pesan kesalahan (*Alternative Flow A3*) untuk melakukan validasi terhadap model ini dapat melihat pada Tabel 4.18, dan Gambar 4.11.



Gambar 4.11 Activity Diagram Mengelola Data Pasien

4.4.3 Pemodelan aktivitas Mengelola Data Konseling Pasien

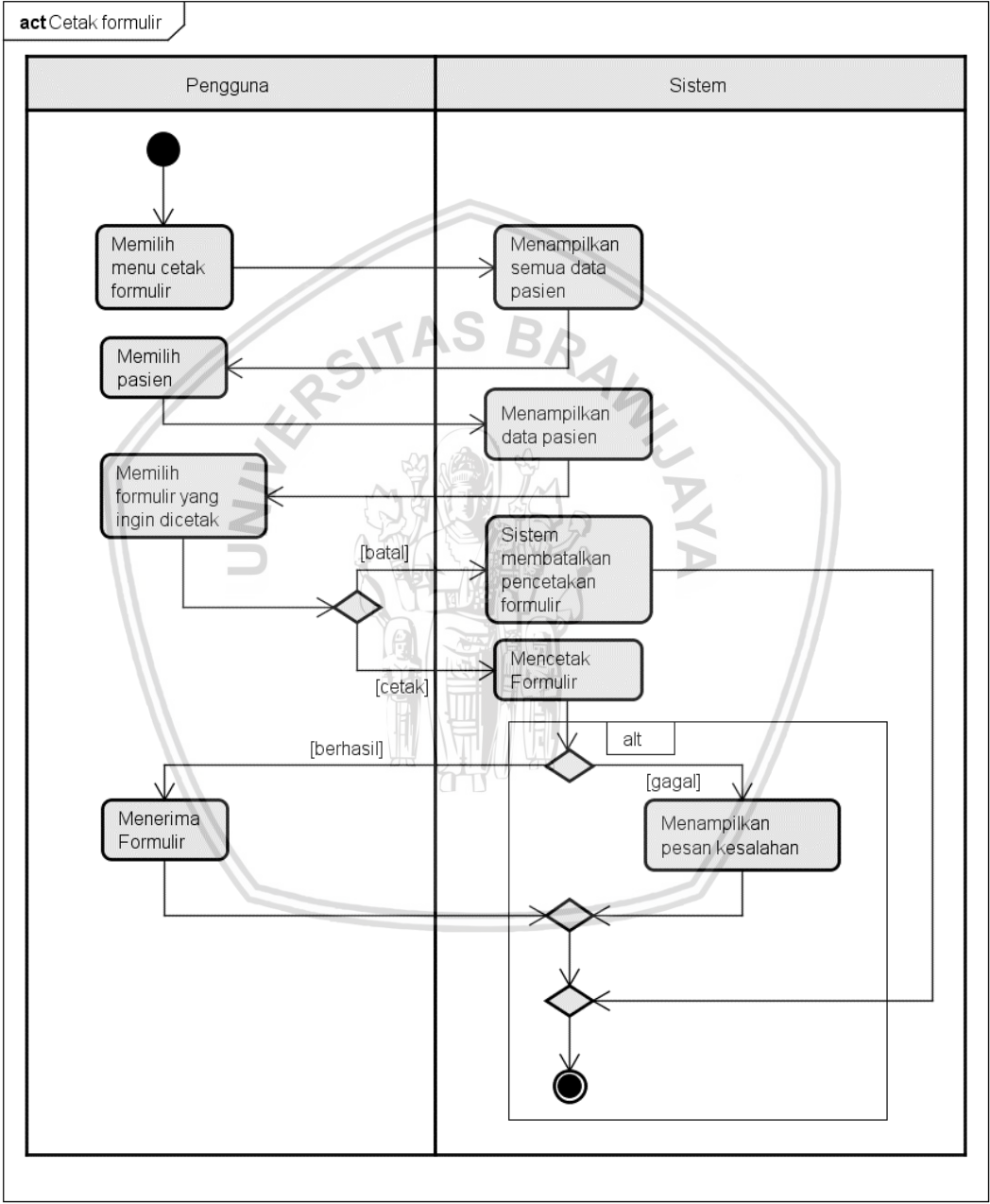
Pemodelan aktivitas mengelola data konseling pasien pertama dilakukan oleh konselor yang memilih menu tersebut, lalu sistem menampilkan data pasien yang telah mempunyai data konseling, lalu konselor memilih pasien yang akan dikelola, selanjutnya memilih dari data konseling yang ada, setelah sistem telah menampilkan data konseling yang dipilih, selanjutnya konselor akan melakukan perubahan dari data konseling tersebut, lalu menyimpannya dan jika terjadi kegagalan menyimpan data maka akan menampilkan pesan kesalahan (*Alternative Flow A2*). Validasi model ini dapat melihat pada Tabel 4.19 dan Gambar 4.12.



Gambar 4.12 Activity Diagram Mengelola Data Konseling Pasien

4.4.4 Pemodelan aktivitas Cetak Formulir

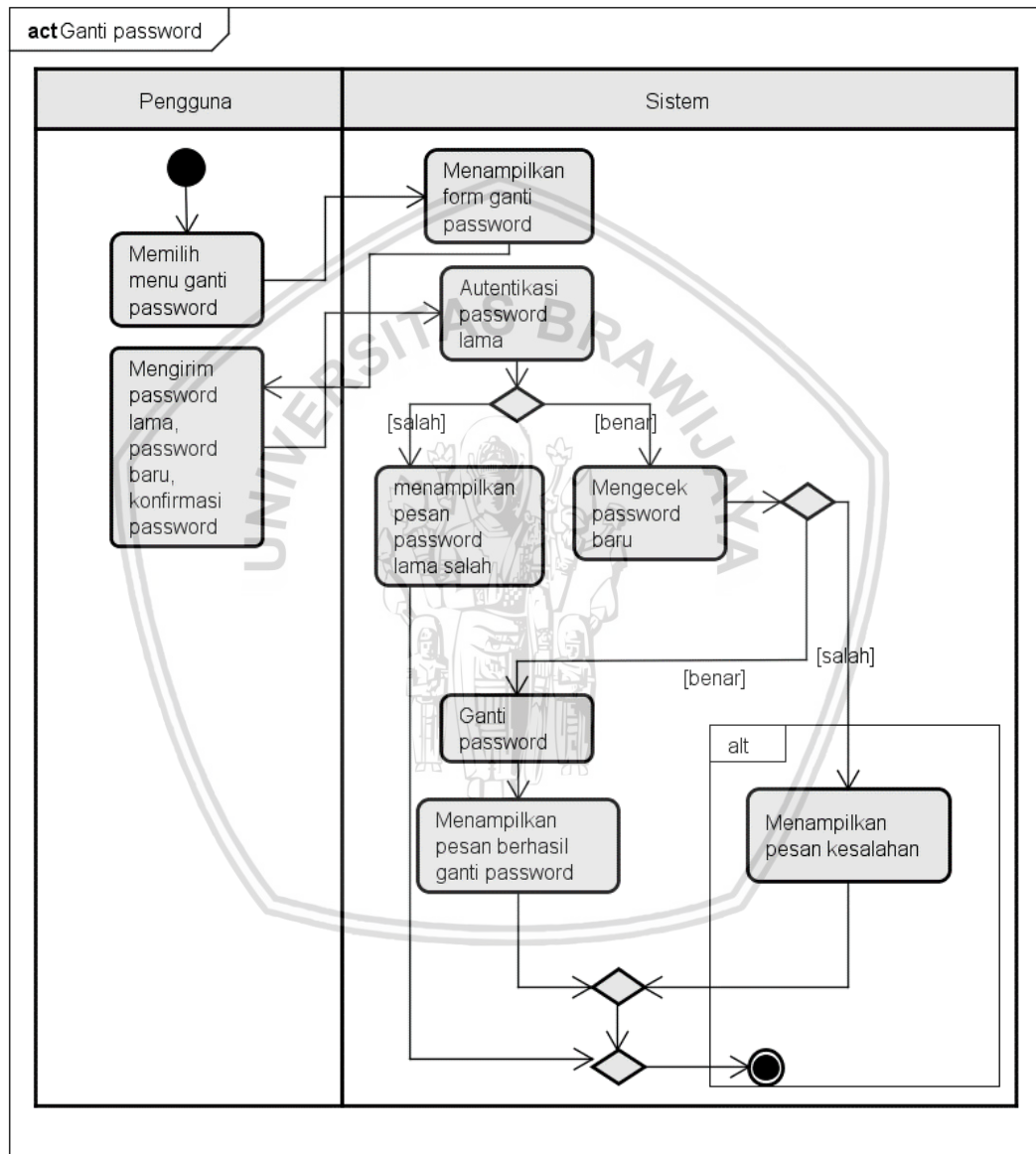
Pemodelan aktivitas cetak formulir dimulai dengan pengguna yang memilih menu cetak formulir, lalu sistem menampilkan semua data pasien sehingga pengguna dapat memilih pasien yang mana akan dicetak formulirnya, setelah itu pengguna memilih formulir apa yang akan dicetak. Model ini dapat dibandingkan dengan Tabel 4.20.



Gambar 4.13 Activity Diagram Cetak Formulir

4.4.5 Pemodelan aktivitas Ganti Password

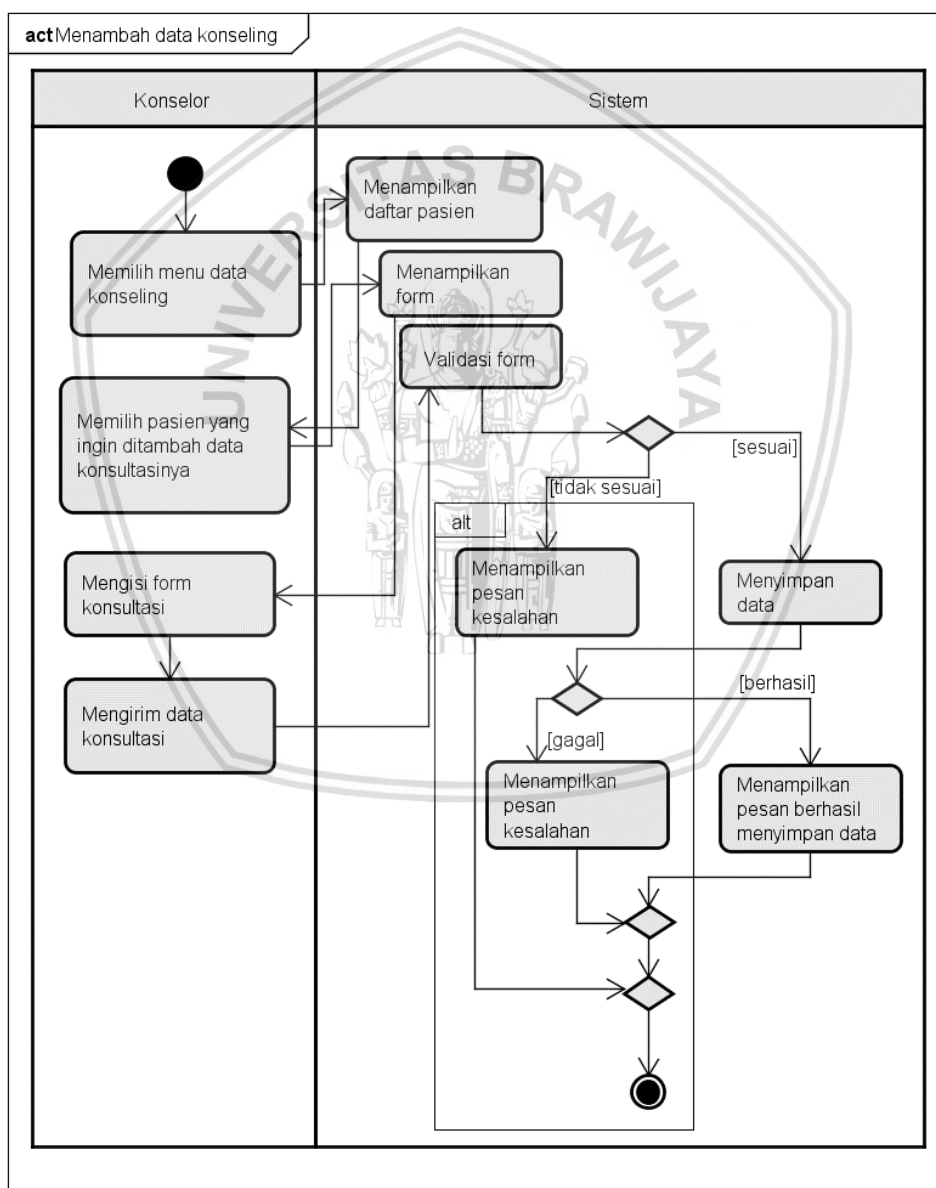
Pemodelan aktivitas ganti *password* dimulai oleh pengguna yang memilih menu mengganti *password*, lalu sistem menampilkan *form* untuk mengisi *password* lama dan *password* baru beserta konfirmasinya, pengguna mengisi lalu mengirimnya. Beberapa *Alternative Flow* dapat dilihat di Tabel 4.21 lalu dibandingkan dengan model ini, lalu sistem akan menampilkan pesan-pesan kesalahan jika *password* lama salah dan kesalahan-kesalahan lainnya.



Gambar 4.14 Activity Diagram Ganti Password

4.4.6 Pemodelan aktivitas Menambah Data Konseling

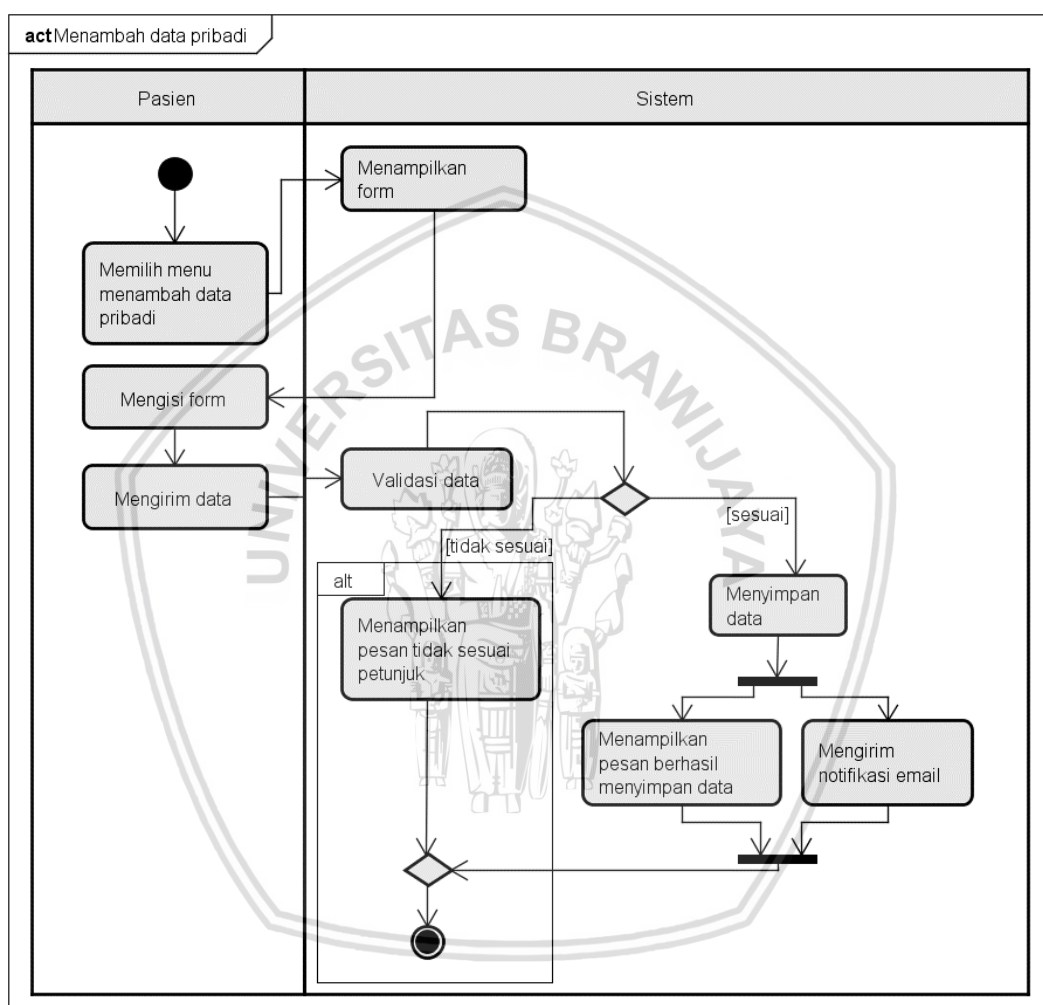
Pemodelan aktivitas ini dimulai dari konselor yang memilih menu menambah data konseling, lalu sistem menampilkan data pasien yang belum mempunyai data konseling, penjelasan mempunyai data konseling adalah yang pasien yang belum konseling sebanyak delapan kali, karena di klinik rehabilitasi BNN Kota Malang mengharuskan pasien melakukan konseling sebanyak delapan kali, maka jika pasien belum delapan kali datanya akan ditampilkan, lalu konselor memilih pasien mana yang akan ditambah data konselingnya, setelah itu menampilkan *form* untuk memasukkan data tersebut. Jika gagal menyimpan data maka akan menampilkan pesan kesalahan (*Alternative Flow A3*), untuk membandingkannya dapat melihat Tabel 4.22.



Gambar 4.15 Activity Diagram Menambah Data Konseling

4.4.7 Pemodelan aktivitas Menambah Data Pribadi

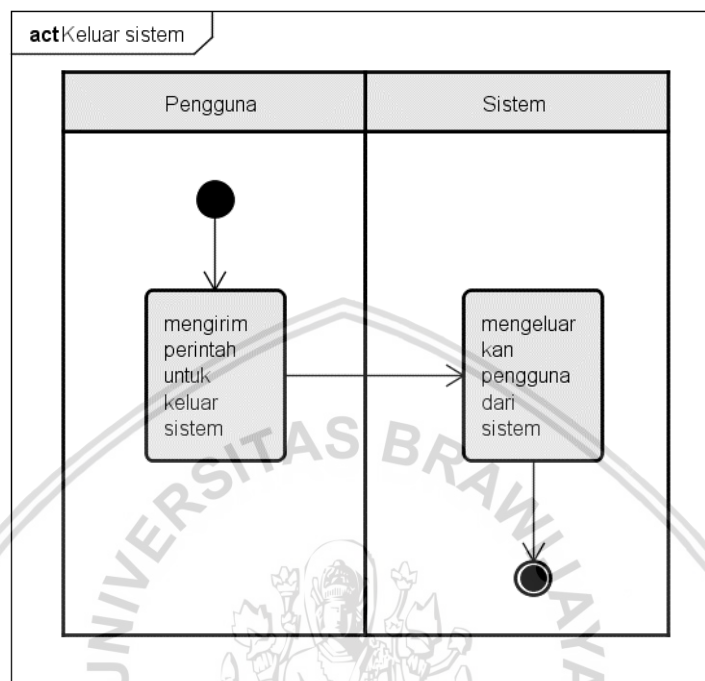
Pemodelan aktivitas ini dimulai dengan pasien yang ingin mendaftar lalu mengakses menu data pribadi, sistem menampilkan *form* lalu pasien mengisi sesuai petunjuk lalu mengirimnya. Jika terjadi kesalahan dalam mengisi *form* maka pesan kesalahan akan ditunjukkan oleh sistem (*Alternative Flow A1*) dapat dilihat di Tabel 4.23. Lalu dilanjutkan dengan sistem mengirim konfirmasi ke email pengguna.



Gambar 4.16 Activity Diagram Menambah Data Pribadi

4.4.8 Pemodelan aktivitas Keluar Sistem

Pemodelan aktivitas keluar sistem dimulai dengan pengguna yang mengirim perintah/memilih menu untuk keluar dari sistem, lalu sistem mengeluarkannya. (Tabel 4.24).



Gambar 4.17 Activity Diagram Keluar Sistem

BAB 5 PERANCANGAN

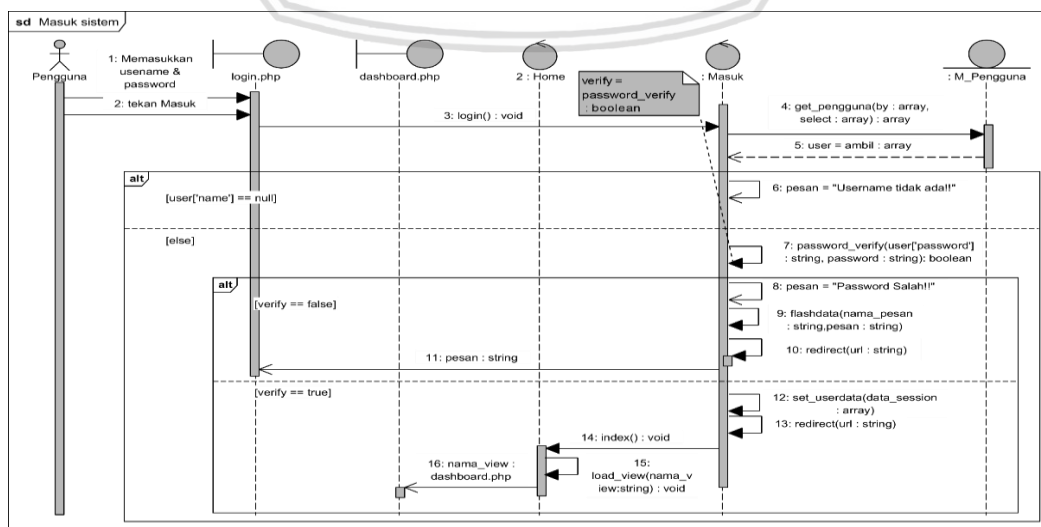
Pada bab ini berisi tentang pembahasan hasil pemodelan menggunakan UML, pada tahap ini terdiri dari pemodelan interaksi antar entitas yaitu *sequence diagram*, lalu pemodelan struktur dan hubungan sistem yaitu diagram kelas, perancangan basis data menggunakan *Physical Data Model*, perancangan antarmuka pengguna, analisis iterasi & inkremental dan perancangan kasus uji.

5.1 Pemodelan Interaksi

Subbab ini akan menjelaskan dan menampilkan pemodelan interaksi yang bersifat *dynamic-behavioural* dengan *sequence diagram* yang dapat menampilkan pertukaran pesan antar entitas aktor, entitas *boundary*, *controller* dan model, pemodelan interaksi ini berdasarkan *use case* yang sudah dibahas pada bab sebelumnya, alur dari *sequence diagram* juga berdasarkan alur dari spesifikasi *use case* dan pemodelan aktivitas, antara lain *sequence diagram* Menampilkan daftar pasien, Menampilkan detail pasien, Mengubah data pasien mengacu pada *use case* Mengelola data pasien, *sequence diagram* Menampilkan daftar pasien konseling, Menampilkan detail konseling, Menambah data konseling, Mengubah data konseling mengacu pada *use case* Mengelola data konseling pasien.

5.1.1 Sequence Diagram Masuk Sistem

Sequence diagram ini menunjukkan bagaimana pengguna dapat teridentifikasi dan masuk ke dalam sistem. Pertama pengguna melengkapi *username* beserta *password* pada form masukkan yang disediakan, lalu menekan tombol *login*, selanjutnya entitas *login* memanggil *function login* yang ada pada *controller* Masuk, lalu dilakukan identifikasi pengguna. Alternatif *fragment* jika *username* tidak ada atau *username* ada tetapi *password* salah, jika benar maka akan mengalihkan ke *controller* Home, akhirnya *controller* Home memuat tampilan yaitu *dashboard.php* yang berarti pengguna sudah berhasil masuk ke dalam sistem.



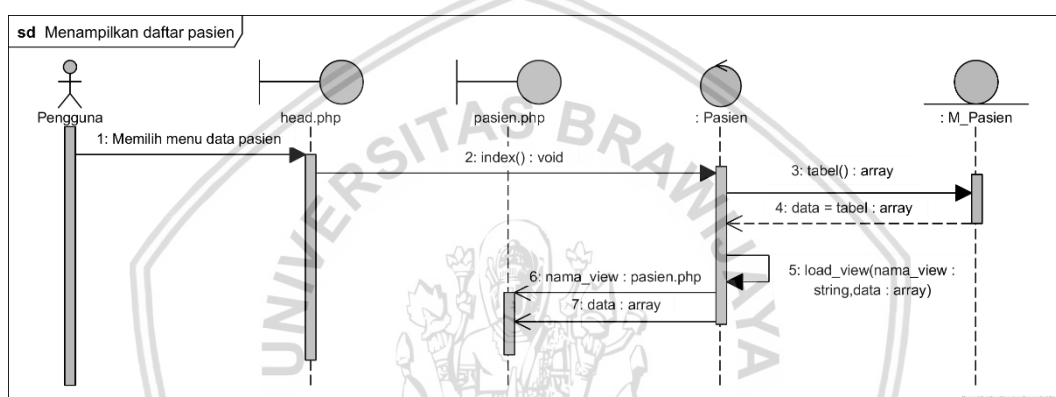
Gambar 5.1 Sequence Diagram Masuk Sistem

5.1.2 Mengelola data pasien

Subbab ini berisi beberapa *sequence* diagram bagaimana aktor petugas dan konselor dapat mengelola data pasien, antara lain menampilkan daftar pasien, melihat detail pasien dan mengubah data pasien, beberapa *sequence* diagram ini menunjukkan bagaimana pertukaran pesan antar objek terjadi dan menunjukkan objek-objek apa yang saling terlibat.

5.1.2.1 Sequence Diagram Menampilkan daftar pasien

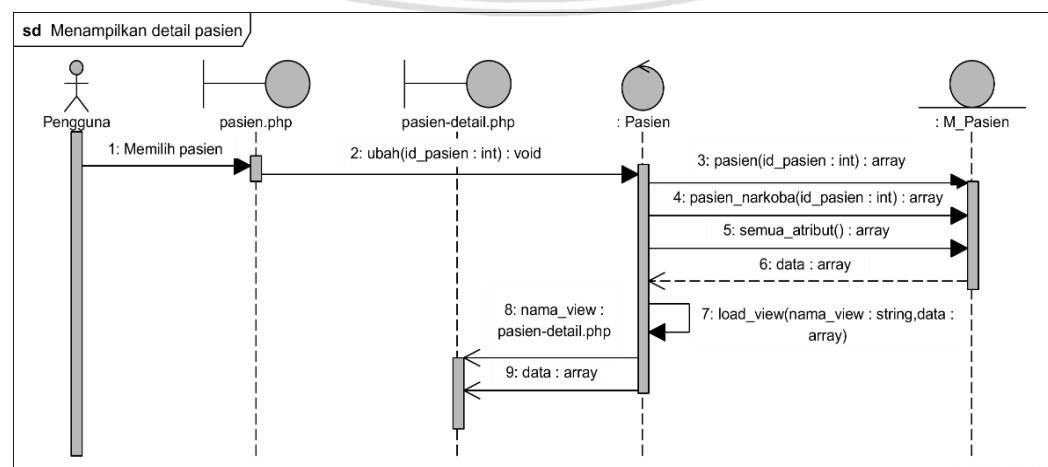
Sequence diagram ini menampilkan bagaimana antar entitas saling bertukar pesan untuk menampilkan daftar pasien, *controller* yang terlibat adalah Pasien lalu entitas *boundary* yang terlibat adalah head.php, dan model yang terlibat adalah M_Pasien.



Gambar 5.2 Sequence Diagram Menampilkan daftar pasien

5.1.2.2 Sequence Diagram Menampilkan detail pasien

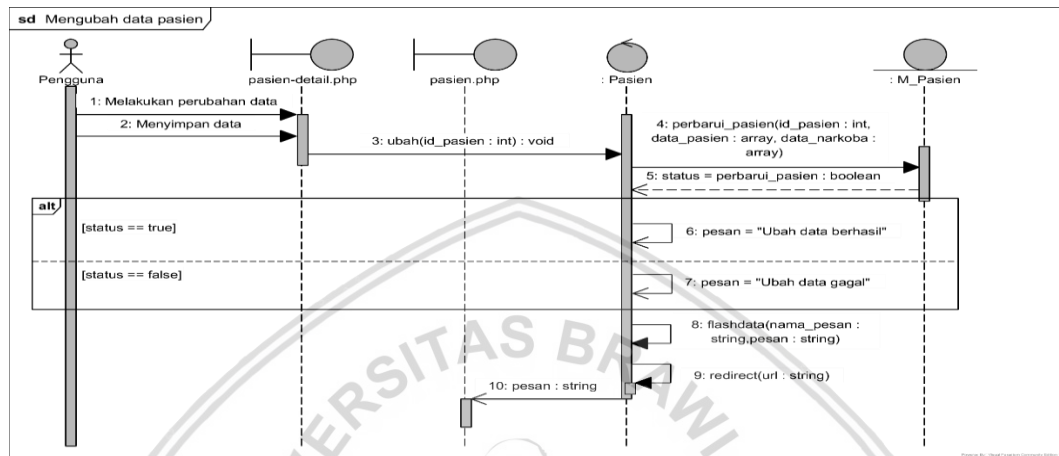
Sequence diagram ini menampilkan bagaimana antar entitas saling bertukar pesan untuk menampilkan detail pasien, *controller* yang terlibat adalah Pasien lalu entitas *boundary* yang terlibat adalah pasien.php dan pasien-detail.php, sedangkan model yang terlibat adalah M_Pasien. Pengguna memulai dari *boundary* pasien.php dan menampilkan detail pada pasien-detail.php.



Gambar 5.3 Sequence Diagram Menampilkan detail pasien

5.1.2.3 Sequence Diagram Mengubah data pasien

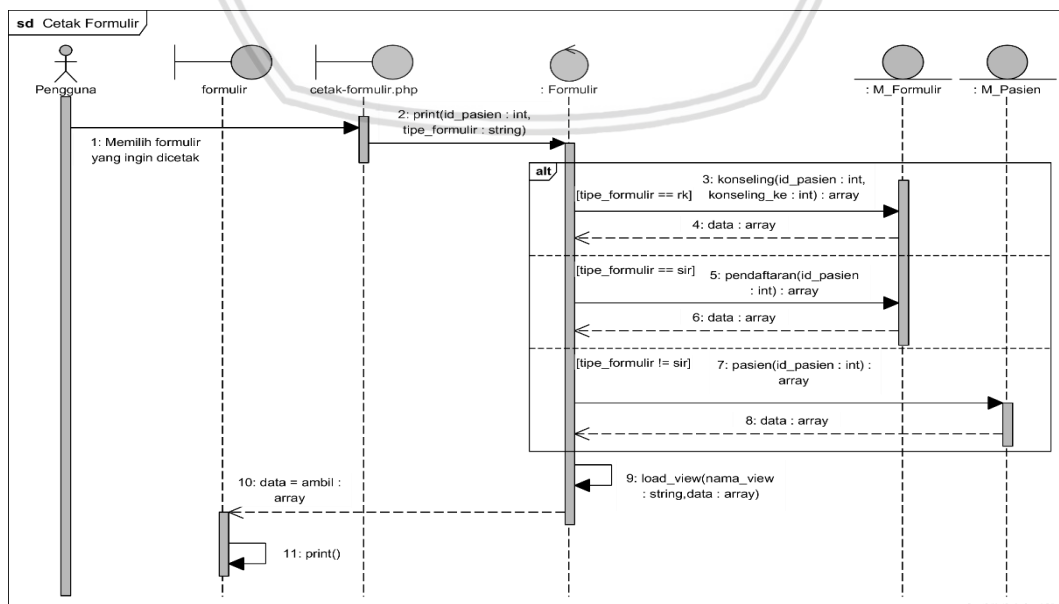
Sequence diagram ini menampilkan bagaimana antar entitas saling bertukar pesan untuk mengubah data pasien, *controller* yang terlibat adalah Home lalu entitas *boundary* yang terlibat adalah pasien-detail.php dan pasien.php, model yang terlibat adalah M_Pasien. Terdapat dua alternatif *flow* jika *reply message* dari M_Pasien bernilai *true* atau *false*, lalu menghasilkan pesan seperti pada Gambar 5.4, pesan tersebut dikirimkan ke *boundary* pasien.php.



Gambar 5.4 Sequence Diagram Mengubah data pasien

5.1.3 Sequence Diagram Cetak formulir

Sequence diagram ini menampilkan bagaimana aktor dapat mencetak formulir, entitas yang terlibat adalah *controller* Formulir, dua model yaitu M_Formulir dan M_Pasien, dan 2 entitas *boundary* (formulir, cetak-formulir.php), formulir berisi form yang akan dicetak sedangkan cetak-formulir.php berisi tombol untuk mencetak form. Tiga *alternatif flow* dengan tipe_formulir menjadi *operand-nya* ditunjukkan pada Gambar 5.5.



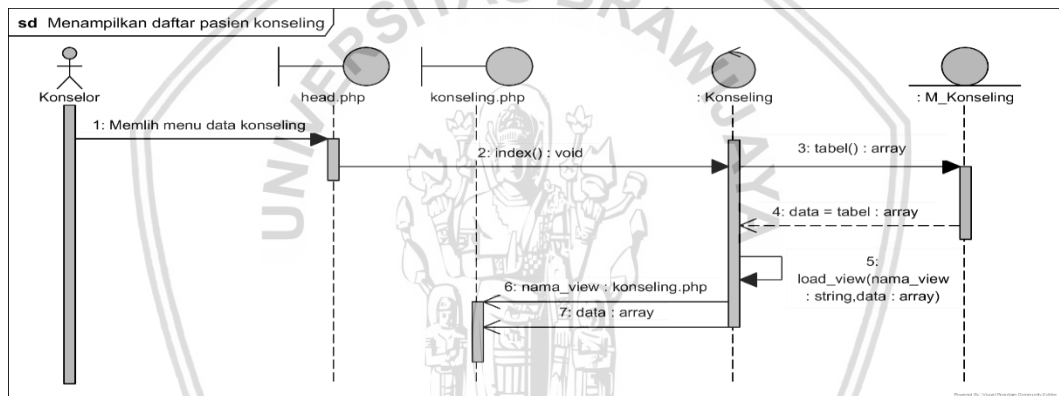
Gambar 5.5 Sequence Diagram Cetak Formulir

5.1.4 Mengelola data konseling pasien

Subbab ini menampilkan beberapa *sequence diagram* yang berkaitan dengan mengelola data konseling pasien, yaitu menampilkan daftar pasien konseling, menampilkan detail konseling dari pasien, menambahkan data konseling dan mengubah data konseling dimana hanya konselor yang bisa melihat dan mengubah data konseling.

5.1.4.1 Sequence Diagram Menampilkan daftar pasien konseling

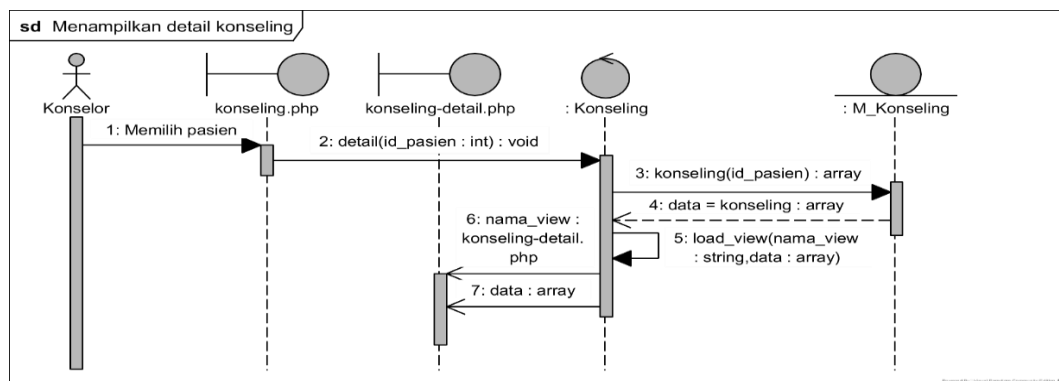
Sequence diagram ini menampilkan bagaimana antar entitas saling bertukar pesan untuk menampilkan daftar pasien konseling, *controller* yang terlibat adalah Konseling lalu entitas *boundary* yang terlibat adalah head.php dan konseling.php, lalu model yang terlibat adalah M_Konseling. Head.php menjadi tampilan untuk pengguna berinteraksi lalu memanggil fungsi index.php pada *controller* Konseling yang selanjutnya memanggil fungsi tabel pada model M_Konseling lalu nilai kembalian dari fungsi tersebut dikirim ke konseling.php setelah mengirimkan tampilan (konseling.php).



Gambar 5.6 Sequence Diagram Menampilkan daftar pasien konseling

5.1.4.2 Sequence Diagram Menampilkan detail konseling

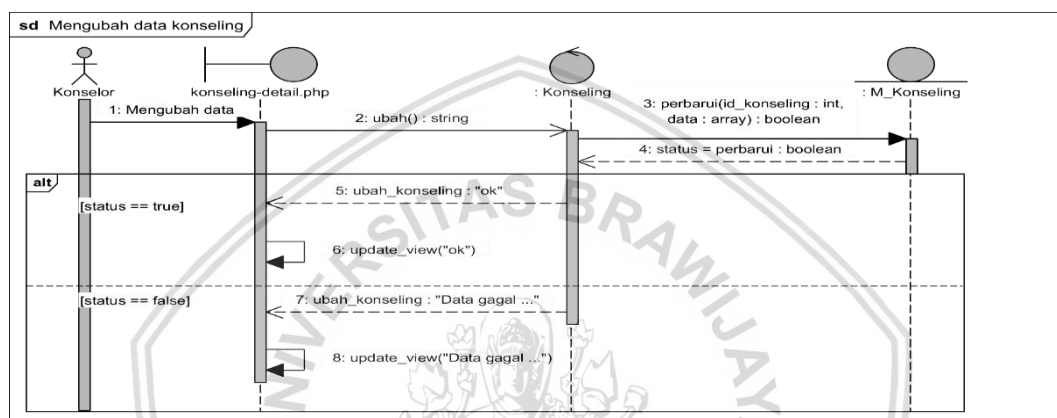
Sequence diagram ini menampilkan bagaimana antar entitas saling bertukar pesan untuk menampilkan detail konseling, *controller* yang terlibat adalah Konseling lalu entitas *boundary* yang terlibat adalah konseling.php dan konseling-detail.php, lalu model yang terlibat adalah M_Konseling.



Gambar 5.7 Sequence Diagram Menampilkan detail konseling

5.1.4.3 Sequence Diagram Mengubah data konseling

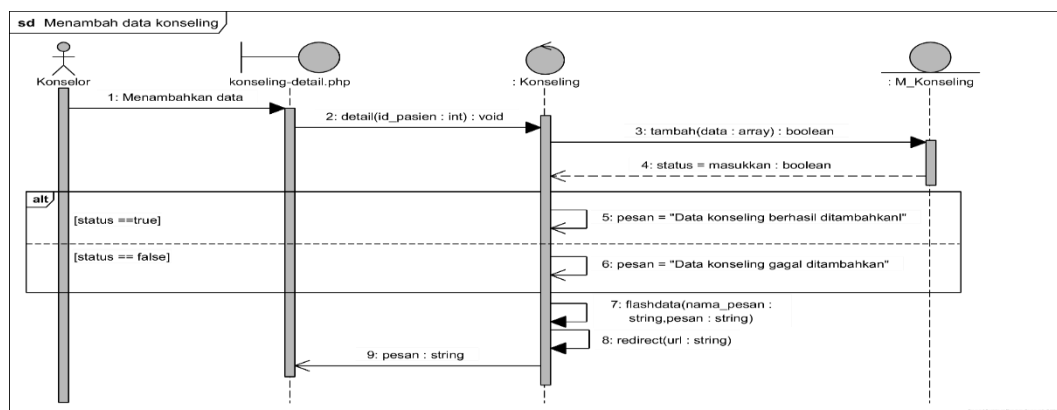
Sequence diagram ini menampilkan bagaimana antar entitas saling bertukar pesan dan memanggil fungsi untuk mengubah data konseling, *controller* yang terlibat adalah Konseling lalu entitas *boundary* yang terlibat adalah konseling-detail.php dan model yang terlibat adalah M_Konseling. Terdapat 2 alternatif *flow* jika status (nilai kembalian dari fungsi perbarui pada model M_Konseling) bernilai true atau false dapat dilihat pada Gambar 5.8, fungsi ubah pada controller Konseling mengembalikan nilai lalu mengirimkannya ke boundary konseling-detail.php setelah itu boundary view memperbarui tampilan seperti pada Gambar 5.8.



Gambar 5.8 Sequence Diagram Mengubah data konseling

5.1.5 Sequence Diagram Menambah data konseling

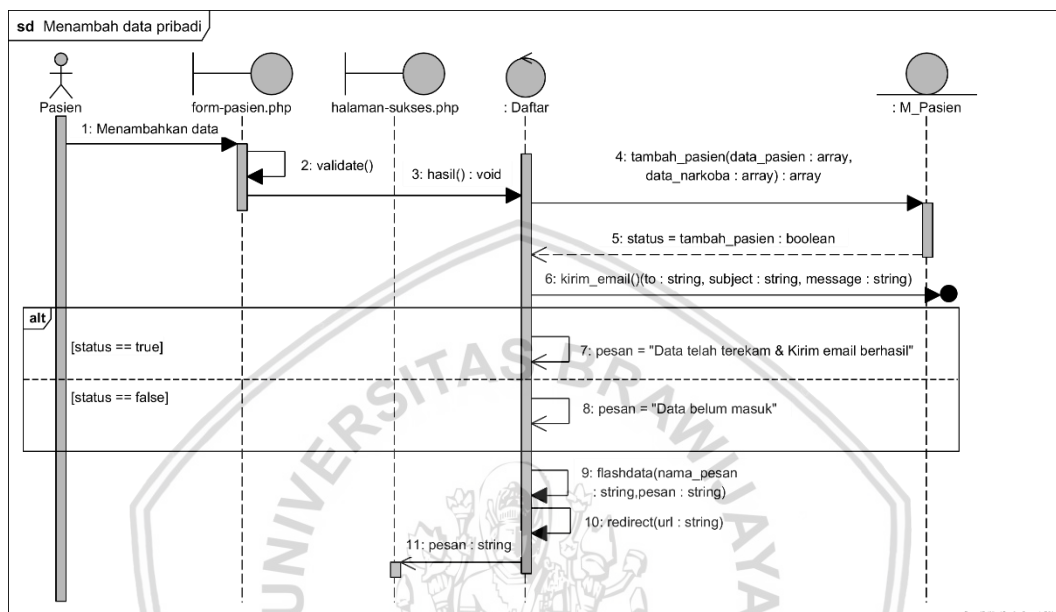
Sequence diagram ini menunjukkan bagaimana antar entitas saling bertukar pesan dan memanggil fungsi untuk menambah data konseling, *controller* yang terlibat adalah Home lalu entitas *boundary* yang terlibat adalah konseling-detail.php dan model yang terlibat adalah M_Konseling. Terdapat dua alternatif jika status (nilai kembalian fungsi tambah pada model M_Konseling) bernilai *true* atau *false*, selanjutnya *controller* Konseling mengirimkan pesan sebagai berikut (Gambar 5.9). lalu mengirimnya ke *boundary* konseling-detail.php.



Gambar 5.9 Sequence Diagram Menambah data konseling

5.1.6 Sequence Diagram Menambah data pribadi

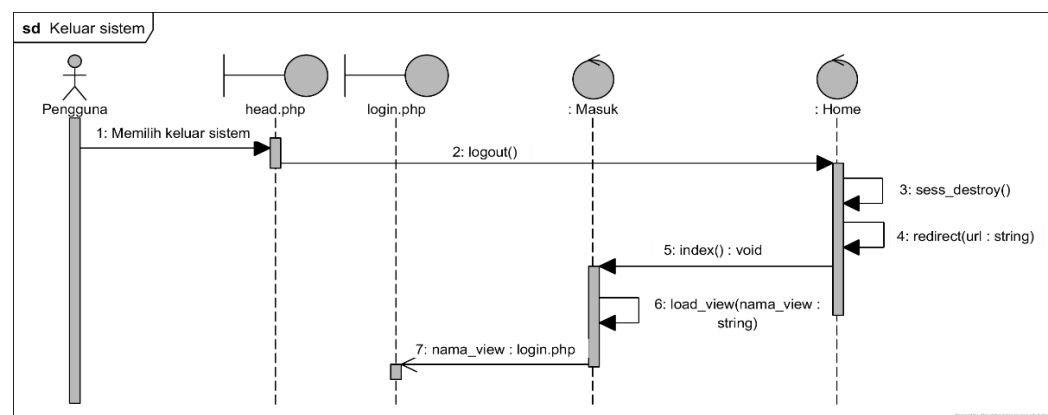
Sequence diagram ini menampilkan bagaimana antar entitas saling bertukar pesan dan memanggil fungsi untuk menambah data pribadi, aktor yang terlibat adalah Pasien, *controller* yang terlibat adalah Daftar lalu entitas *boundary* yang terlibat adalah form-pasien.php dan halaman-sukses.php, lalu model yang terlibat adalah M_Pasien.



Gambar 5.10 Sequence Diagram Menambah data pribadi

5.1.7 Sequence Diagram Keluar Sistem

Sequence diagram ini menampilkan bagaimana alur pesan untuk dapat keluar dari sistem, aktor yang terlibat adalah Pengguna, *controller* yang terlibat adalah Home dan Masuk lalu entitas *boundary* yang terlibat adalah head.php dan login.php, sedangkan tidak ada model yang terlibat. Pengguna berinteraksi dengan *boundary* head.php yang memanggil fungsi logout pada controller Home, setelah menghapus *session* selanjutnya redirect ke controller Masuk lalu memuat tampilan login.php.



Gambar 5.11 Sequence Diagram Keluar Sistem

5.2 Pemodelan Objek

Pemodelan objek merupakan cara menggambarkan bagaimana struktur hubungan dari suatu sistem yang akan dikembangkan, pemodelan objek menggunakan diagram kelas. Diagram kelas analisis merupakan panduan untuk tahap berikutnya yaitu merancang basis data dan diagram kelas perancangan, sedangkan diagram kelas perancangan merupakan panduan untuk mengembangkan sistem yang didapatkan dari pola MVC dalam *sequence diagram*.

5.2.1 Diagram Kelas Analisis

Diagram kelas analisis disusun dari analisis penulis terhadap objek yang akan dikembangkan nantinya pada sistem, hubungan asosiasi dan generalisasi juga menggambarkan bagaimana hubungan tiap kelas. Diagram kelas Pengguna merupakan representasi objek yang menyimpan atribut sebagai berikut dan mempunyai beberapa operasi yang merupakan kelas induk dari kelas petugas, dan kelas petugas merupakan kelas induk dari kelas konselor.

Kelas pengguna memiliki hubungan asosiasi *many-to-one* terhadap kelas Data_Rehab, yang berarti banyak objek pengguna mempunyai hanya 1 objek Data_Rehab, dan kelas Data_Rehab mempunyai atribut kumpulan objek pasien dan membuat asosiasi *one-to-many* terhadap objek pasien. Kelas konselor merupakan kelas turunan dari petugas mempunyai atribut Konseling yang juga dimiliki oleh pasien dan memunculkan hubungan asosiasi *one-to-many*. Kelas konseling merupakan representasi objek yang menyimpan data terkait konseling.

Kelas pasien memiliki beberapa atribut yang memunculkan hubungan asosiasi, pada kelas diagram analisis ini kelas pasien yang merupakan representasi objek yang menyimpan atribut sebagai berikut mempunyai hubungan asosiasi - *one-to-one* terhadap kelas Sumber_Pasien, Sumber_Biaya, dan Provinsi, sedangkan mempunyai hubungan asosiasi *one-to-many* terhadap kelas konseling dan mempunyai hubungan asosiasi *many-to-many* terhadap kelas Narkoba.

5.2.2 Diagram Kelas Perancangan

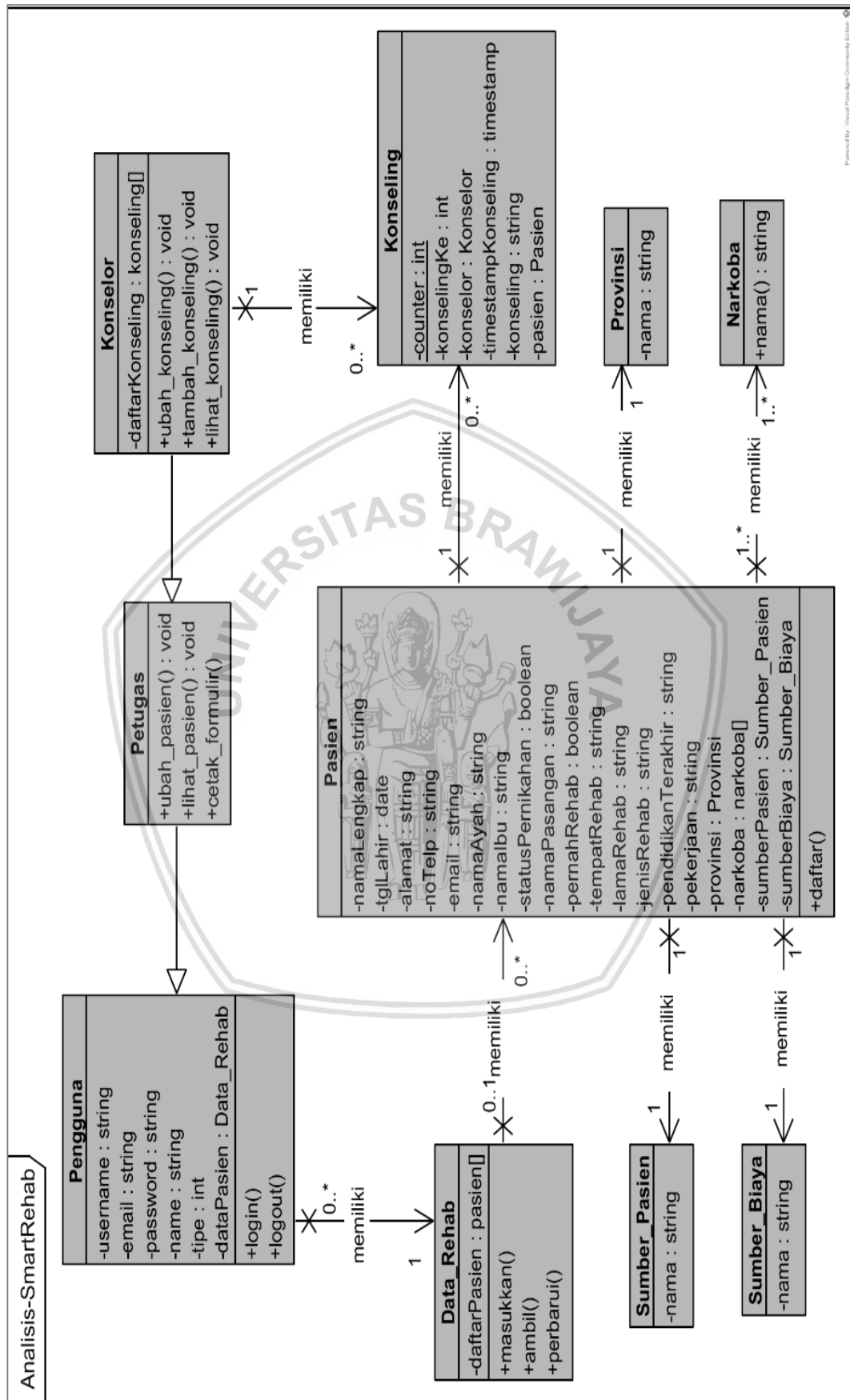
Pengembangan diagram kelas perancangan didapatkan dari diagram kelas analisis (Gambar 5.12) yang disesuaikan dengan pengembangan menggunakan *framework CodeIgniter PHP*, dan kelas-kelas yang teridentifikasi dari *sequence diagram*. Diagram kelas perancangan terdiri dari Diagram Kelas Perancangan *Controller*, Diagram Kelas Perancangan Model, dan Hubungan dari keduanya.

Subkelas dari *CI_Controller* terdiri dari enam kelas (Gambar 5.14) yaitu kelas *Home*, *Masuk* dan *Daftar*, kelas *Home* terdiri dari fungsi (*behaviour*) pengguna sistem, yaitu Petugas dan Konselor, dalam controller *Home* terdapat 4 fungsi yaitu *index* untuk menampilkan halaman dashboard, *ganti_password* untuk mengganti password, *notif* untuk memberikan notifikasi lalu *logout* untuk keluar dari sistem. *Controller Masuk* berisi dua fungsi untuk menampilkan halaman login (*index*) dan melakukan otentifikasi dan identifikasi pengguna (*login*). *Controller pasien* juga terdiri dari dua fungsi, yang pertama *index* untuk menampilkan tabel

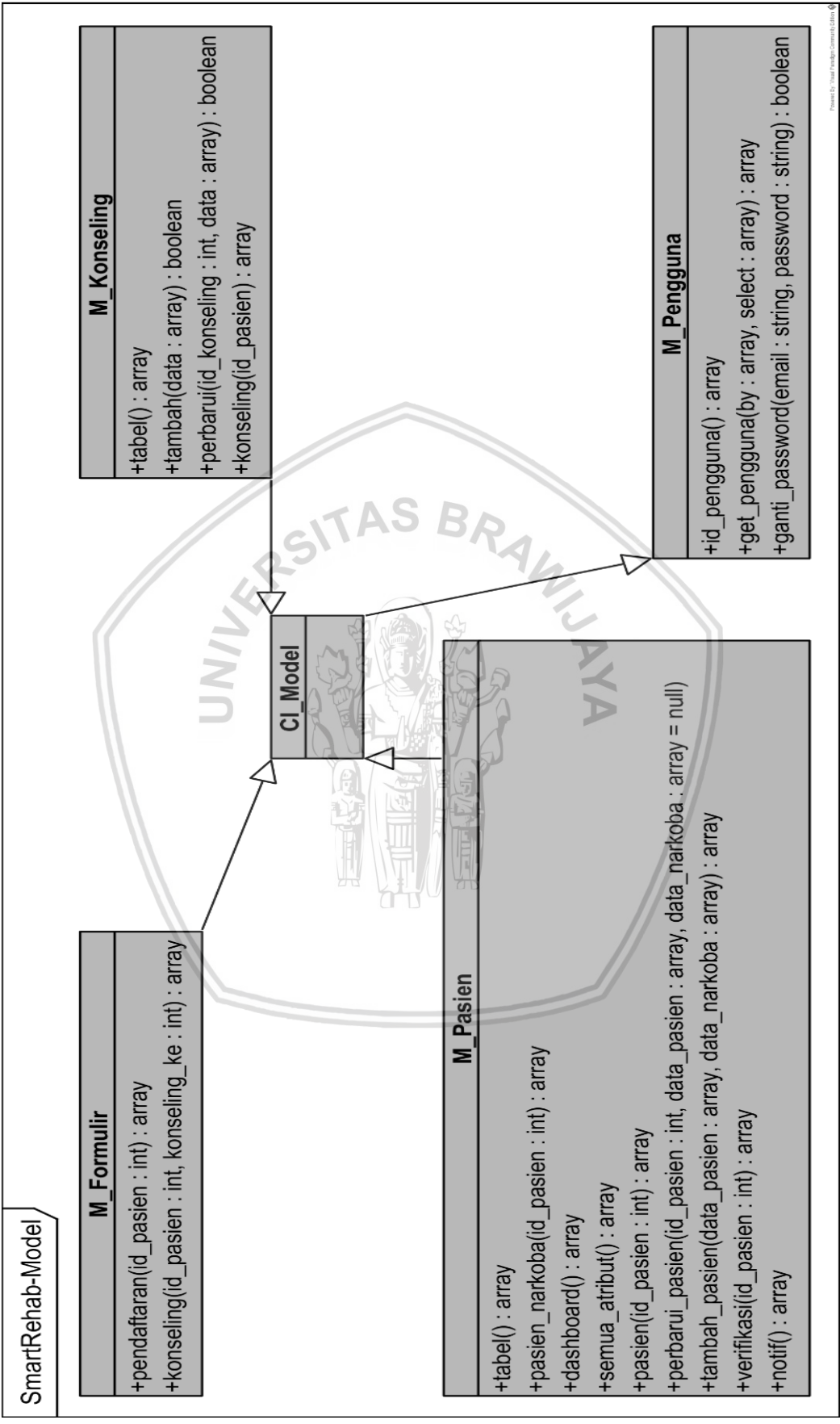
daftar pasien lalu fungsi kedua adalah ubah untuk menampilkan detail pasien dan melakukan perubahan data pada pasien. *Controller* Formulir terdiri dari 3 fungsi, fungsi index untuk menampilkan daftar pasien, fungsi detail untuk menampilkan daftar formulir dari pasien sedangkan fungsi print untuk mencetak formulir. *Controller* Konseling bertugas untuk menangani data terkait konseling, Controller ini terdiri dari 3 fungsi, fungsi index untuk menampilkan daftar pasien, fungsi ubah untuk mengubah data konseling, sedangkan fungsi detail untuk menampilkan detail dari data konseling pasien. *Controller* Daftar menangani pendaftaran pasien controller ini terdiri dari 3 fungsi, fungsi index untuk menampilkan formulir pendaftaran, fungsi hasil untuk menangkap masukkan dari pasien yang mengisi formulir lalu menyimpan ke database melalui model M_Pasien, satu fungsi lagi adalah kirim_email dengan *visibility private* bertujuan untuk mengirimkan konfirmasi email. Fungsi-fungsi diatas merupakan kebutuhan fungsional yang diperoleh dari analisis persyaratan pada bab 4.

Sedangkan Subkelas dari *CI_Model* terdiri dari empat kelas (Gambar 5.13) Model M_Pasien terdiri dari beberapa fungsi yang bertugas mengakses data dari database, M_Pasien menangani data-data terkait data pasien dari mengambil data memperbarui data dan menambah data, model M_Pengguna terdiri dari empat fungsi yang berguna menangani data terkait pengguna sistem yaitu mengganti password dan login. Model M_Konseling bertugas untuk menangani data terkait konseling, fungsi terdiri dari tambah data, perbarui data dan mengambil data, sedangkan model M_Formulir bertugas untuk menangani data terkait formulir, data yang diambil dari tabel Pasien yang dibutuhkan oleh formulir.

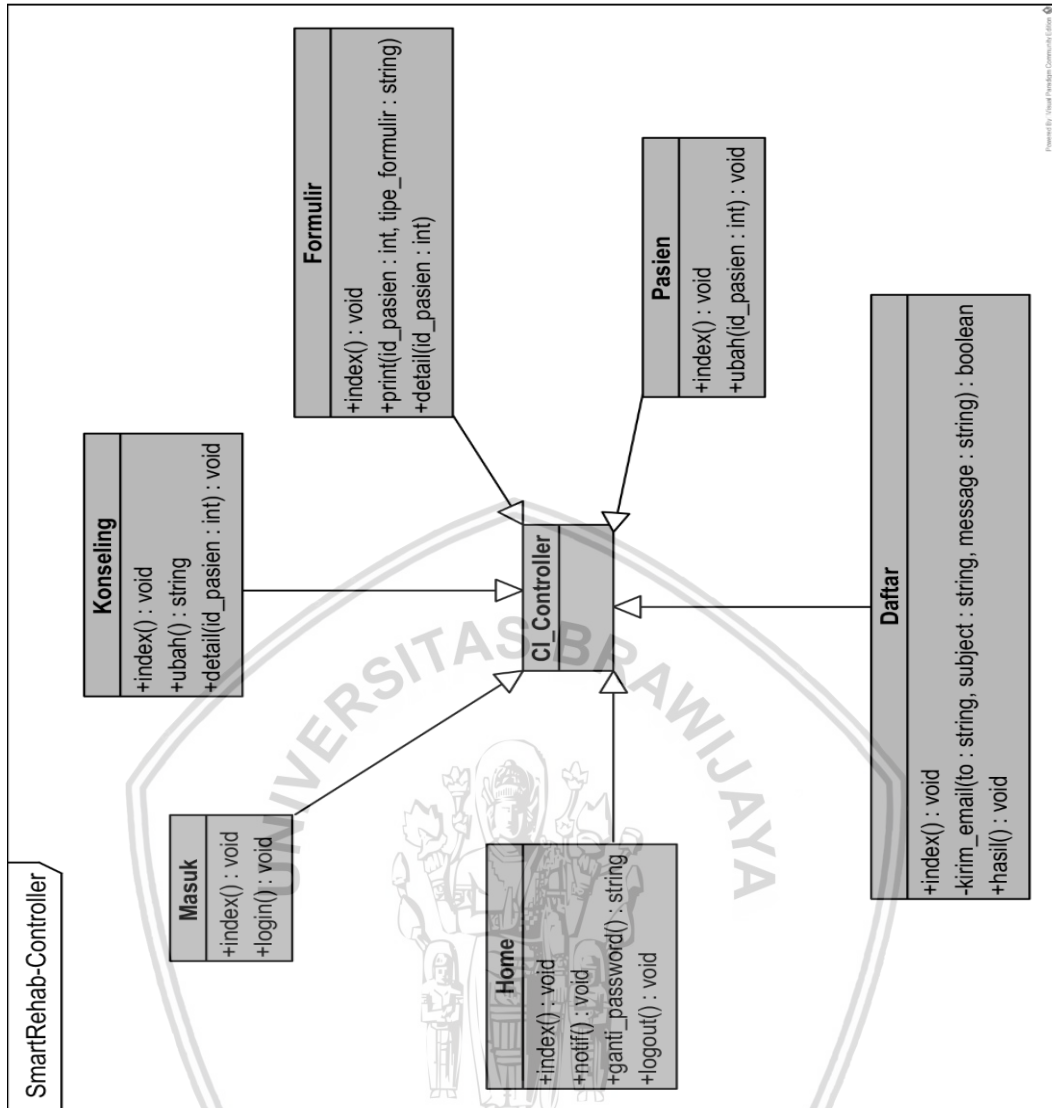
Hubungan dari controller dan model pada Gambar 5.15 menunjukkan controller menggunakan model apa saja, Controller terdiri dari kumpulan logika dari sistem ini sedangkan model berisi akses data ke tabel dalam *database*. Model-model ini merupakan representasi dari tabel pada *database*.



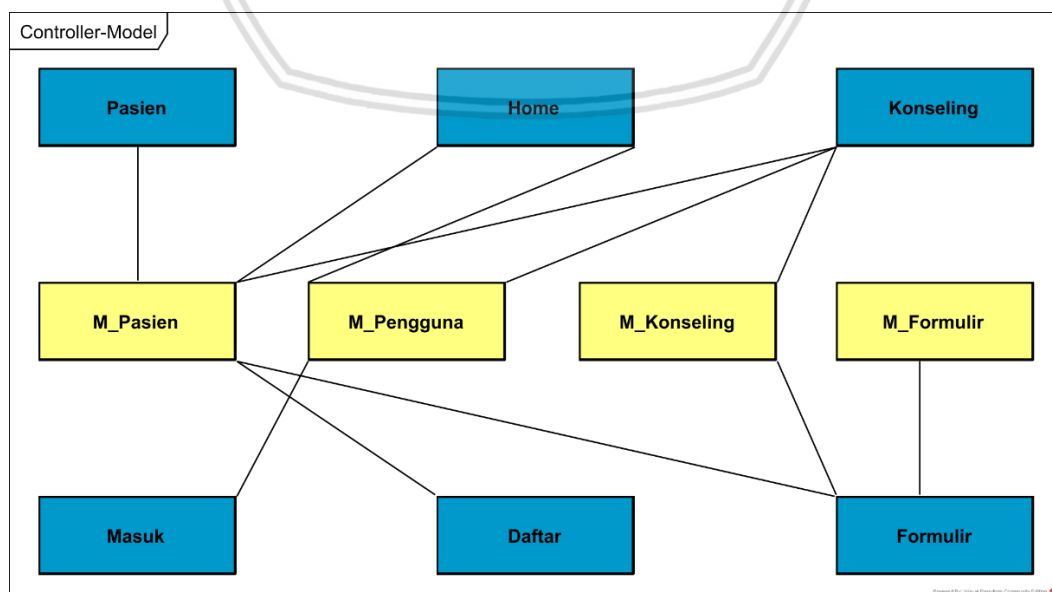
Gambar 5.12 Diagram Kelas Analisis



Gambar 5.13 Diagram Kelas Perancangan Model



Gambar 5.14 Diagram Kelas Perancangan *Controller*



Gambar 5.15 Diagram Kelas Perancangan Relasi *Controller & Model*

5.4 Perancangan Algoritme

Tahap ini merupakan tahap mendokumentasikan sekumpulan perintah, operasi logika sebagai konsep pembangunan dari setiap kebutuhan fungsional dengan *pseudocode* yang nantinya akan menjadi panduan dalam mengembangkan sistem pada tahap implementasi.

5.4.1 Masuk Sistem

Algoritme dimulai dengan mengambil data hasil masukkan dari pengguna, lalu mengambil data dari *database* sesuai masukkan pengguna, jika tersedia maka sistem akan mengalihkan ke *dashboard*, dan jika tidak akan mengalihkan ke tampilan *login* dengan pesan kesalahan.

Tabel 5.1 Pseudocode Masuk Sistem

No	Pseudocode
1	START
2	READ data from input then assign to input_user
3	READ data from database where = input_user
4	IF EXIST
5	SET session_data
6	REDIRECT to dashboard
7	ELSE
8	REDIRECT to login_view show error message
9	ENDIF
10	END

5.4.2 Mengelola data pasien

Algoritme dimulai dengan mengambil masukkan data yang merupakan hasil perubahan data dalam mengelola data pasien, lalu mengubah data di dalam *database* menggunakan data masukkan perubahan data pasien, jika perubahan berhasil akan menampilkan pesan berhasil, jika tidak maka menampilkan pesan kesalahan.

Tabel 5.2 Pseudocode Mengelola Data Pasien

No	Pseudocode
1	START
2	READ data from input then assign to data_pasien_array
3	UPDATE data from data_pasien_array to database
4	IF SUCCESS
5	SHOW success message
6	ELSE
7	SHOW error message
8	ENDIF
9	END

5.4.3 Cetak formulir

Algoritme dimulai dengan mengambil masukkan data dari pengguna berupa tipe formulir dan id pasien, Lalu mengambil data dari *database* berdasarkan id pasien dan memuat data berdasarkan tipe formulir yang diberikan. Setelah memuat formulir perintah selanjutnya adalah mencetak formulir.

Tabel 5.3 Pseudocode Cetak Formulir

No	Pseudocode
1	START
2	READ data from input then assign to tipe_formulir
3	READ data from input then assign to id_pasien
4	READ data from database where id_pasien = id_pasien then assign to data_pasien
5	IF tipe_formulir = 'dpn'
6	LOAD dpn_view then assign data from data_pasien
7	PRINT dpn_view
8	ENDIF
9	IF tipe_formulir = 'sir'
10	LOAD sir_view then assign data from data_pasien
11	PRINT sir_view
12	ENDIF
13	IF tipe_formulir = 'lpt'
14	LOAD lpt_view then assign data from data_pasien
15	PRINT lpt_view
16	ENDIF
17	IF tipe_formulir = 'lpr'
18	LOAD lpr_view then assign data from data_pasien
19	PRINT lpr_view
20	ENDIF
21	IF tipe_formulir = 'ak'
22	LOAD ak_view then assign data from data_pasien
23	PRINT ak_view
24	ENDIF
25	END

5.4.4 Mengelola data konseling pasien

Algoritme dimulai dengan mengambil data dari masukkan pengguna yang berupa perubahan data konseling pasien lalu data tersebut menjadi masukkan untuk melakukan perubahan pada *database*, jika data berhasil diubah maka akan menampilkan pesan sukses sebaliknya jika gagal akan menampilkan pesan kesalahan.

Tabel 5.4 Pseudocode Mengelola data konseling pasien

No	Pseudocode
1	START
2	READ data from input then assign to data_konseling_array
3	UPDATE data from data_konseling_array to database
4	IF SUCCESS
5	SHOW success message
6	ELSE
7	SHOW error message
8	ENDIF
9	END

5.4.5 Menambah data konseling pasien

Algoritme untuk menambah data konseling adalah pertama mengambil data dari masukkan pengguna, setelah itu data tersebut dimasukkan ke dalam variabel yang nantinya akan dimasukkan ke dalam *database*, pesan akan ditampilkan jika data tersebut berhasil atau gagal dimasukkan ke dalam *database*.

Tabel 5.5 Pseudocode Menambah data konseling pasien

No	Pseudocode
1	START
2	READ data from input then assign to data_konseling_array
3	INSERT data from data_konseling_array to database
4	IF SUCCESS
5	SHOW success message
6	ELSE
7	SHOW error message
8	ENDIF
9	END

5.4.6 Menambahkan data pribadi

Algoritme ini dijalankan oleh calon pasien, pertama mengambil data dari masukkan pengguna setelah memasukkan data tersebut ke dalam *database*, pesan akan ditampilkan apakah data pasien tersebut berhasil ditambahkan, jika berhasil tidak hanya pesan ditampilkan tetapi akan dikirimkan konfirmasi melalui email ke alamat email calon pasien yang telah dicantumkan.

Tabel 5.6 Pseudocode Menambahkan data pribadi

No	Pseudocode
1	START
2	READ data from input then assign to data_pasien_array
3	INSERT data from data_pasien_array to database
4	IF SUCCESS
5	SHOW success message
6	SEND email confirmation
7	ELSE
8	SHOW error message
9	ENDIF
10	END

5.4.7 Ganti *password*

Algoritme ini hanya dijalankan oleh pengguna sistem yaitu konselor dan petugas, yang pertama mengambil data dari masukkan pengguna yaitu email, *password* lama, *password* baru beserta konfirmasinya, jika *password* lama tidak sama maka akan muncul pesan kesalahan begitu juga jika *password* lama sama dengan *password* baru akan muncul pesan kesalahan. Pesan kesalahan akan muncul jika *password* konfirmasi tidak sama setelah lolos semua seleksi kondisi maka akan dilakukan perubahan *password*, jika tidak berhasil dalam mengubah *password* akan menampilkan pesan kesalahan, dan jika berhasil pesan berhasil akan ditampilkan.

Tabel 5.7 Pseudocode Ganti Password

No	Pseudocode
1	START
2	READ data from input then assign to email_input
3	READ data from input then assign to password_lama_input
4	READ data from input then assign to password_baru
5	READ data from input then assign to password_konfirmasi
6	READ data from database where = id_pasien then assign to password_lama
7	READ data from database where = id_pasien then assign to email

Tabel 5.7 Pseudocode Ganti Password (Lanjutan)

8	IF email_input = email
9	IF password_lama_input = password_lama
10	IF password_baru = password_konfirmasi
11	IF password_baru = password_lama
12	SHOW error message
13	ELSE
14	UPDATE data password_baru to database
15	IF SUCCESS
16	SHOW success message
17	ELSE
18	SHOW error message
19	ENDIF
20	ENDIF
21	ELSE
22	SHOW error message
23	ENDIF
24	ELSE
25	SHOW error message
26	ENDIF
27	ELSE
28	SHOW error message
29	ENDIF
30	END

5.4.8 Keluar Sistem

Algoritme ini untuk mengeluarkan pengguna dari sistem dengan cara meng-*unset* (hapus) *session* data yang dimiliki pengguna saat masuk sistem.

Tabel 5.8 Pseudocode Keluar Sistem

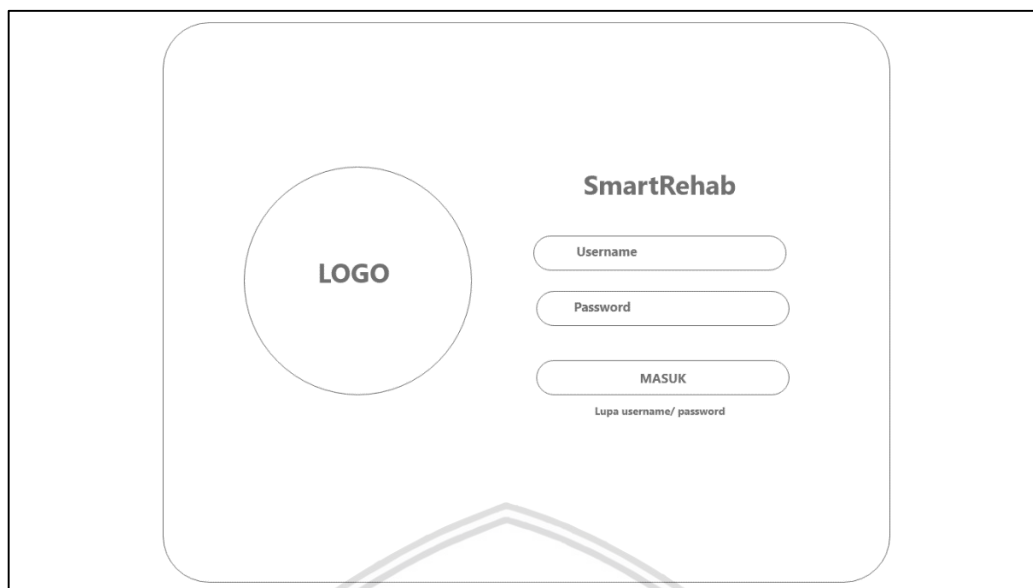
No	Pseudocode
1	START
2	UNSET session_data
3	REDIRECT to login_view
4	END

5.5 Perancangan Antarmuka Pengguna

Bagian ini berisi sketsa yang merupakan rancangan tampilan antarmuka yang akan dikembangkan nantinya, terdiri dari halaman masuk sistem, halaman menu mengelola data pasien, mengelola data konseling, ganti *password*, cetak formulir dan sketsa tampilan bagi pasien untuk menambahkan data pribadi. Dalam pengembangan nantinya penulis menggunakan *framework* *css bootstrap* yang telah banyak digunakan, yang menjadi alasan penggunaan *framework* ini adalah penggunaannya yang dapat mempercepat proses pengembangan antarmuka sistem informasi ini karena komponen sudah dibuat dan hanya me-*reuse* dalam penggunaannya.

5.5.1 Antarmuka Masuk sistem

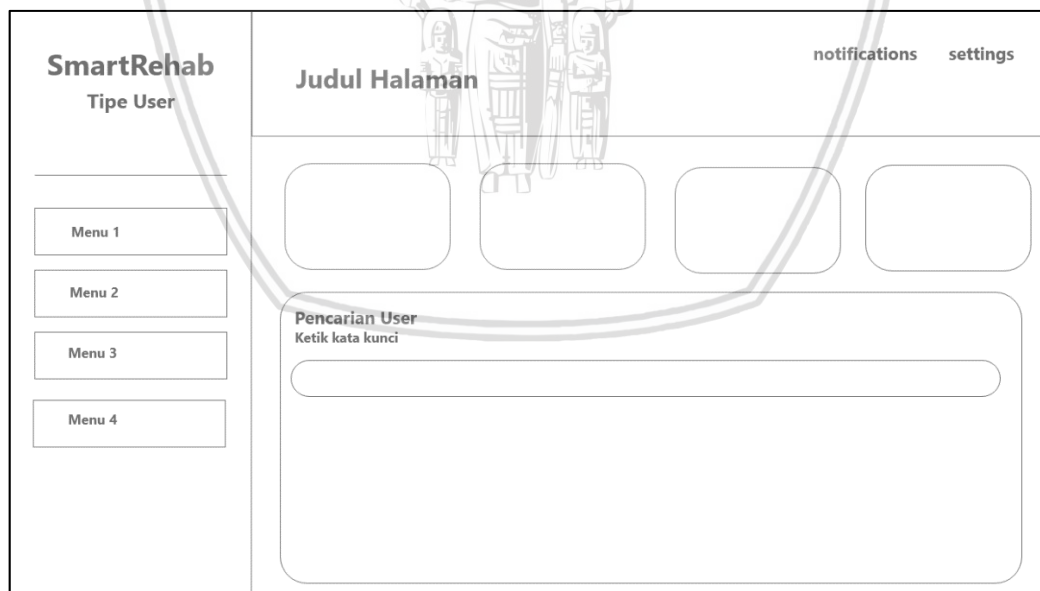
Pada gambar di bawah ini merupakan antarmuka yang akan dikembangkan untuk masuk ke sistem, logo di sebelah kiri dan *form* untuk memasukkan *username* dan *password* beserta tombol untuk masuk, pada rancangan antarmuka ini disediakan jarak antar masukkan dan tombol untuk memberikan informasi terkait kegagalan pada proses masuk sistem.



Gambar 5.17 Sketsa Tampilan Antarmuka Masuk Sistem

5.5.2 Antarmuka *Dashboard*

Antarmuka *dashboard* merupakan tampilan jika berhasil masuk ke dalam sistem, berisi beberapa “card” yang berisi informasi terkini terkait sistem informasi data pasien, di sebelah kiri tersedia *sidebar* untuk memudahkan navigasi antar menu dalam sistem informasi di pojok kanan atas terdapat dua menu tambahan yaitu notifikasi dan pengaturan untuk mengganti *password* atau keluar sistem.



Gambar 5.18 Sketsa Tampilan Antarmuka Dashboard

5.5.3 Antarmuka Mengelola data pasien

Gambar di bawah ini merupakan antarmuka untuk mengelola data pasien, penulis menggunakan tab untuk membagi masukkan menjadi tiga bagian untuk mempermudah pengubahan/penulisan data dan meningkatkan ketelitian dalam

mengubah data karena tidak secara langsung disodorkan banyak *input* dalam satu halaman.

Gambar 5.19 Sketsa Tampilan Antarmuka Mengelola data pasien

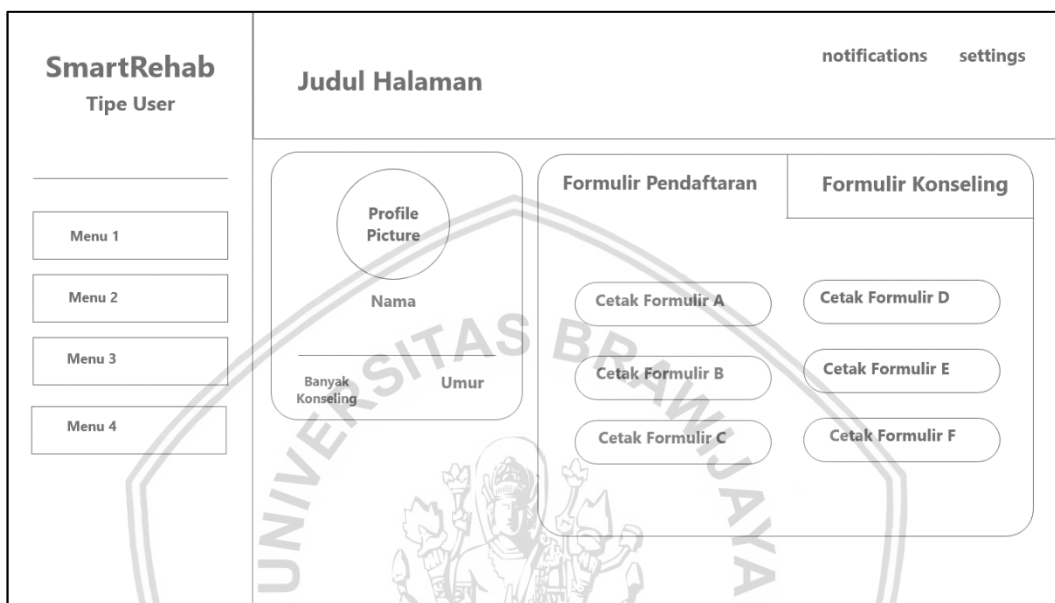
5.5.4 Antarmuka Mengelola data konseling pasien

Antarmuka ini berbeda dengan tampilan untuk mengelola data pasien, karena dalam rancangan tampilan ini terdapat *section* untuk menampilkan *profile picture* (jika pasien mengunggahnya), umur dan banyak konseling, lalu di bawahnya menampilkan data konseling terdahulu dari pasien, dimana pengguna bisa mengubah atau menghapusnya, di sebelah kanan terdapat *textarea* untuk memasukkan data konseling dan langsung menyimpannya, *textarea* di sebelah kiri akan muncul jika jumlah konseling dari pasien tersebut di bawah delapan, karena sesuai dengan aturan yang berlaku satu pasien hanya bisa konseling sebanyak delapan kali.

Gambar 5.20 Sketsa Tampilan Antarmuka Mengelola data konseling pasien

5.5.5 Antarmuka Cetak formulir

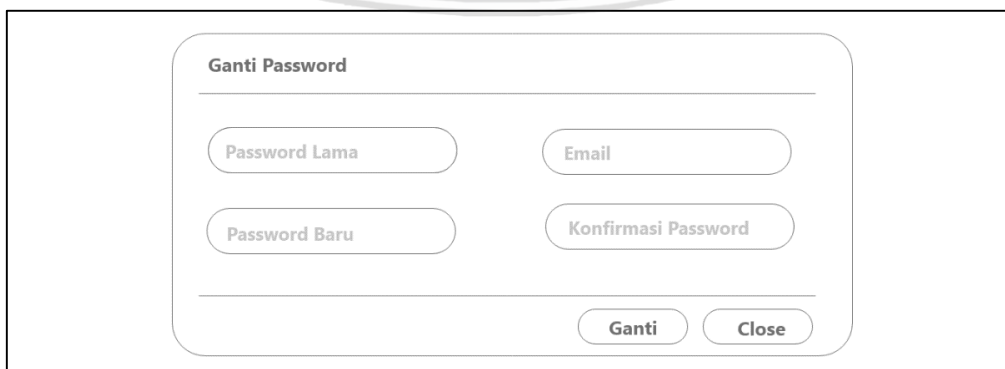
Sketsa antarmuka untuk cetak formulir hampir sama seperti data konseling pasien, karena di sebelah kiri terdapat “card” yang berisi *profile picture*, nama, banyak konseling dan umur, tetapi di sebelah kanan ada 2 tab yang berisi beberapa tombol untuk mencetak formulir, untuk tipe akun petugas hanya bisa mencetak pada formulir yang tersedia pada tab formulir pendaftaran, lalu untuk tipe akun konselor dapat mencetak semua formulir.



Gambar 5.21 Sketsa Tampilan Antarmuka Cetak Formulir

5.5.6 Antarmuka Ganti *password*

Sketsa antarmuka untuk ganti *password* adalah dengan menampilkan “modal” yang berisi masukkan sebagai berikut (Gambar 5.19), tampilan “modal” dipilih karena lebih ringkas dan cepat (tidak perlu pindah halaman) untuk mengganti *password* dapat langsung meng-klik menu tambahan di sebelah kanan atas.



Gambar 5.22 Sketsa Tampilan Antarmuka Ganti Password

5.5.7 Antarmuka Menambahkan data pribadi

Sketsa antarmuka untuk menambahkan data pribadi terdiri dari banyak masukan, sehingga penulis menggunakan “wizard” yang terdiri dari tiga tab yang mempunyai pembagian seperti pada fitur mengelola data pasien, lalu disediakan fitur untuk mem-validasi masukan dari *user*, sehingga *form* lebih intuitif dan memastikan masukan dari pasien benar dan valid.

Gambar 5.23 Sketsa Tampilan Antarmuka Menambahkan data pribadi

5.6 Analisis Inkremental & Iterasi

Subbab ini membahas penggunaan strategi iterasi dan inkremental dalam pengembangan sistem informasi pasien rehabilitasi di BNN Kota Malang (SmartRehab). Pertama adalah memisahkan implementasi sistem menjadi beberapa bagian (strategi inkremental), peneliti menggunakan strategi dekomposisi berdasarkan fungsional karena dengan strategi ini pembagian dari pembangunan sistem terlihat jelas dan dapat diukur, estimasi iterasi dan kesulitan dari inkremen berdasarkan kemampuan dari peneliti dan, banyaknya komponen yang dibangun, dan kompleksitas dari fungsi itu sendiri, sedangkan prioritas dan *dependency* berdasarkan hasil wawancara validasi masalah.

Tabel 5.9 Analisis Inkremental & Iterasi Pengembangan SmartRehab

Inkremen	Komponen	Estimasi Iterasi	Prioritas	Kesulitan	Keterkaitan
Masuk Sistem	<ul style="list-style-type: none"> Antarmuka pengguna (Halaman <i>login</i>) 	2-3	Tinggi	Sedang	-

Tabel 5.9 Analisis Inkremental & Iterasi Pengembangan SmartRehab (Lanjutan)

	<ul style="list-style-type: none"> • <i>Function</i> pada <i>Controller</i> • <i>Library</i> untuk <i>hashing</i> • <i>Session</i> untuk membedakan tipe <i>user</i> • Tabel pada <i>database</i> 				
Keluar Sistem	<ul style="list-style-type: none"> • <i>Function</i> pada <i>controller</i> 	1			Masuk Sistem
Ganti <i>Password</i>	<ul style="list-style-type: none"> • Antarmuka pengguna (Modal) • <i>Function</i> pada <i>controller</i> • AJAX pada <i>frontend</i> 	2-3	Sedang	Sedang	Masuk Sistem
Menambah data pribadi	<ul style="list-style-type: none"> • Antarmuka pengguna (Halaman pendaftaran) • Validasi masukkan pengguna di <i>frontend</i> menggunakan <i>library jquery-validate</i> • Penggunaan <i>library bootstrap-select</i> untuk menyimpan masukkan pengguna yang terdiri dari <i>multiple option</i> 	4-5	Tinggi	Tinggi	-

Tabel 5.9 Analisis Inkremental & Iterasi Pengembangan SmartRehab (Lanjutan)

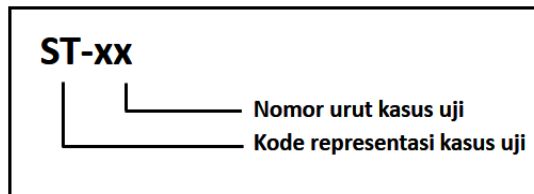
	<ul style="list-style-type: none"> • <i>Function</i> pada <i>controller</i> • <i>Function</i> pada <i>model</i> • Perancangan tabel pasien pada <i>database</i> • Penggunaan <i>library</i> email untuk mengirimkan konfirmasi email 				
Mengelola data pasien	<ul style="list-style-type: none"> • Antarmuka pengguna (Halaman daftar pasien) • Penggunaan <i>library</i> datatables (jQuery) • Notifikasi pada <i>frontend</i> menggunakan AJAX • <i>Function</i> pada <i>controller</i> • <i>Function</i> pada <i>model</i> untuk meng-<i>update</i> 	3-4	Tinggi	Sedang	Masuk Sistem, Menambah data pribadi
Menambah data konseling	<ul style="list-style-type: none"> • Antarmuka pengguna (Halaman daftar pasien) • Penggunaan <i>library</i> datatables (jQuery) • <i>Function</i> pada <i>controller</i> 	3-4	Sedang	Mudah	Mengelola data pasien

Tabel 5.9 Analisis Inkremental & Iterasi Pengembangan SmartRehab (Lanjutan)

	<ul style="list-style-type: none"> • Perancangan tabel konseling 				
Mengubah data konseling	<ul style="list-style-type: none"> • Antarmuka pengguna (Halaman daftar konseling) • Penggunaan modal dan AJAX di <i>frontend</i> • <i>Function</i> pada <i>controller</i> 	3-4	Sedang	Sedang	Menambah data konseling
Cetak Formulir	<ul style="list-style-type: none"> • Antarmuka pengguna (Halaman daftar formulir) • Pembuatan dokumen • <i>Function</i> pada <i>controller</i> • <i>Query</i> untuk tiap formulir • <i>Parsing</i> data dari <i>database</i> ke formulir • <i>Function</i> pada <i>frontend</i> untuk mencetak 	4-5	Tinggi	Tinggi	Menambah data pribadi, menambah data konseling

5.7 Perancangan Pengujian

Subbab ini akan menjelaskan bagaimana merancang *test-case* beserta metode testing apa yang akan dipakai pada bab 7 (pengujian). Peneliti menggunakan metode *scenario testing* yang diperkenalkan oleh (Kaner, 2003a) yang merupakan salah satu dari *black-box* testing, sedangkan untuk tabel perancangan pengujian mengadaptasi dari buku dari (Agarwal et al., 2010), tujuannya dari subbab ini adalah merancang kasus-kasus uji pada bab 7.



Gambar 5.24 Kodefikasi Kasus Uji

5.7.1 Perancangan Pengujian Skenario

Pemilihan kasus-kasus uji berdasarkan hasil wawancara validasi masalah dimana *stakeholder* mengemukakan cerita-cerita bagaimana kesulitan mereka dalam melakukan suatu aktivitas, penggunaan metode pengujian ini diharapkan tidak hanya menjawab kesulitan dari *stakeholder* tetapi juga dapat secara tidak langsung mengajar calon pengguna, menurut (Kaner, 2003b) *scenario* pada *test-case* pengujian *scenario* testing bisa diambil dari *use case scenario* (spesifikasi *use case*), beberapa tabel di bawah merupakan rancangan *test case* yang peneliti rancang sesuai teori dari metode *scenario testing* oleh (Kaner, 2003a).

Beberapa kasus uji peneliti rancang berdasarkan kebutuhan pengguna yang didapatkan dari analisis persyaratan, beberapa skenario nantinya mungkin unik dan jarang dialami oleh pengguna sistem, tetapi diharapkan nantinya sistem dapat menangani jika terjadi skenario seperti itu. Beberapa skenario akan dirancang menggunakan batas waktu yaitu pada skenario ST-01 dan ST-02, dengan pengujian skenario pengujian dapat memberikan beberapa data atau parameter untuk menguji sistem dengan skenario tersebut, skenario ST-02 merupakan skenario dari proses bisnis usulan (*to-be*) Proses Bisnis To-Be Mencari Data Pasien. Kedua skenario tersebut diberikan batas waktu untuk mengetahui apakah pengembangan sistem berhasil seperti yang direncanakan pada tahap analisis persyaratan.

Tabel 5.10 Kasus Uji Skenario Pendaftaran Pasien Baru

Pendaftaran Pasien Baru	ST-01
Tujuan Pengujian	Menguji apakah sistem dapat mendaftarkan pasien baru
Skenario Pengujian	<ol style="list-style-type: none"> 1. Calon pasien mengisi form dan mengumpulkan menggunakan SmartRehab 2. Pasien yang mendaftar sebanyak tiga orang bersamaan, dua orang menggunakan <i>platform mobile</i>, satu orang menggunakan <i>desktop browser</i>

Tabel 5.10 Kasus Uji Skenario Pendaftaran Pasien Baru (Lanjutan)

	3. Petugas melakukan verifikasi lalu mencetak formulir pendaftaran
Hasil yang diharapkan	Formulir tercetak sesuai data dari pasien dan data tersimpan di dalam <i>database</i>
Batas Waktu	5 jam 3 menit 30 detik

Tabel 5.11 Kasus Uji Skenario Pencarian berdasarkan kategori

Pencarian berdasarkan kategori	ST-02
Tujuan Pengujian	Menguji apakah sistem dapat menampilkan hasil pencarian berdasarkan kategori
Skenario Pengujian	<ol style="list-style-type: none"> 1. Petugas membuat <i>query</i> pencarian berdasarkan kategori 2. Petugas mencari data berdasarkan kategori 3. Sistem menampilkan hasil pencarian
Hasil yang diharapkan	Sistem menampilkan hasil pencarian
Batas waktu	21 detik

Tabel 5.12 Kasus Uji Skenario Menambah data konseling

Menambah data konseling	ST-03
Tujuan Pengujian	Menguji apakah sistem dapat menyimpan data konseling
Skenario Pengujian	<ol style="list-style-type: none"> 1. Konselor melakukan konseling terhadap pasien 2. Konselor memasukkan data ke dalam <i>form</i> lalu menyimpannya 3. Konselor mencetak <i>form</i>
Hasil yang diharapkan	Formulir tercetak sesuai data konseling dari pasien dan data konseling tersimpan di dalam <i>database</i>

5.7.2 Pengujian *Usability*

Usability testing merupakan pengujian pengguna dimana yang diuji adalah pengguna setelah menggunakan sistem informasi, tabel di bawah ini berisi kuesioner untuk menguji *usability* dari sistem ini menggunakan kuesioner USE (*Usefulness, Satisfaction dan Ease of Use*), kuesioner ini akan diisi oleh pengguna sistem yaitu petugas/ staf rehabilitasi BNN Kota Malang dan Para pasien yang mendaftar menggunakan sistem informasi ini, menurut Spool & Schroeder (2001) enam sampai delapan responden/pengguna cocok untuk melakukan pengujian *usability*, dan staf rehabilitasi yang hanya berjumlah enam orang, sehingga peneliti merencanakan banyak responden yang diambil adalah enam orang dimana terdiri beberapa staf rehabilitasi dan staf bagian umum BNN Kota Malang.

Tabel 5.13 Rancangan Kuesioner untuk menguji *Usability*

NO	PERNYATAAN	JAWABAN				
		STS	TS	N	S	SS
		1	2	3	4	5
Usefulness						
1	Sistem ini membantu saya dalam mengelola data rehabilitasi menjadi lebih efektif					
2	Sistem ini membantu saya dalam mengelola data rehabilitasi menjadi lebih produktif					
3	Sistem ini sangat berguna bagi saya					
4	Sistem ini memberikan saya penguasaan lebih dalam mengelola data rehabilitasi					
5	Sistem ini mempermudah saya mengelola data rehabilitasi					
6	Sistem ini membuat saya menghemat waktu saat mengelola data rehabilitasi					
7	Sistem ini memenuhi kebutuhan saya dalam mengelola data rehabilitasi					
8	Sistem ini sesuai dengan apa yang saya harapkan					
Easy to Learn						
9	Saya dapat mempelajari menggunakan Sistem ini dengan cepat					
10	Saya mudah mengingat bagaimana menggunakan Sistem ini					
11	Saya dengan mudah menggunakan Sistem ini					

12	Saya dengan cepat menguasai Sistem ini					
<i>Easy to Use</i>						
13	Sistem ini mudah digunakan					
14	Sistem ini dapat digunakan dengan sederhana					
15	Sistem ini dapat digunakan oleh semua kalangan masyarakat					
16	Sistem ini memerlukan langkah-langkah sederhana untuk dapat mencapai apa yang saya inginkan					
17	Sistem ini dapat digunakan secara fleksibel					
18	Tidak membutuhkan banyak usaha untuk menggunakan Sistem ini					
19	Saya dapat menggunakan Sistem ini tanpa instruksi tertulis					
20	Saya tidak menemukan inkonsistensi (ketidakesesuaian) selama menggunakan Sistem ini					
21	Baik pengguna reguler maupun yang bukan reguler, akan menyukai Sistem ini					
22	Saya dapat dengan cepat dan mudah memperbaiki kesalahan saya dalam menggunakan Sistem					
23	Saya dapat menggunakan Sistem ini dengan praktis saat membutuhkannya					
<i>Satisfaction</i>						
24	Saya puas dengan Sistem ini					
25	Saya akan merekomendasikan Sistem ini kepada orang lain					
26	Sistem ini menyenangkan untuk digunakan					
27	Sistem ini bekerja seperti yang saya inginkan					
28	Sistem ini sangat bagus					
29	Saya ingin memiliki Sistem ini					
30	Sistem ini nyaman untuk digunakan					

5.7.3 Pengujian Kompatibilitas

Pengujian ini dirancang untuk memenuhi kebutuhan *stakeholder*, mereka membutuhkan sistem yang dapat dipakai dimana saja, ini berarti sistem harus mampu menjalankan fungsionalitasnya pada browser apa saja termasuk pada platform *mobile*. Peneliti menggunakan *tools* SortSite Trial 5.29.818.0 untuk menguji kompatibilitas sistem informasi ini, *tools* ini menyediakan hampir semua kasus uji untuk browser dan platform *mobile* yang ada hingga saat ini, tabel di bawah merupakan kasus uji yang akan diujikan pada sistem informasi yang dikembangkan.

Tabel 5.14 Perancangan Kasus Uji Kompatibilitas

Platform/Browser	Versi
Internet Explorer	9,10,11
Edge	16
Firefox	60
Safari	< 10,11
Opera	51
Chrome	66
iOS	< 9,10,11
Android	< 3,4
BlackBerry	< 7.1,10

BAB 6 IMPLEMENTASI

Bab implementasi terdiri dari penjelasan mengenai kebutuhan dalam mengimplementasi sistem, implementasi algoritme menjadi kode program dan implementasi antarmuka berdasarkan perancangan dari bab sebelumnya.

6.1 Spesifikasi Lingkungan Implementasi

Subbab ini berisi kebutuhan dalam implementasi sistem informasi, Tabel 6.1 berisi spesifikasi perangkat keras untuk melakukan implementasi, Tabel 6.2 berisi spesifikasi perangkat lunak untuk melakukan implementasi, sedangkan Tabel 6.3 berisi spesifikasi yang direkomendasikan untuk lingkungan instalasi dari sistem informasi ini. Tabel 6.1 dan Tabel 6.2 merupakan spesifikasi dari perangkat yang digunakan oleh peneliti. *Library* yang digunakan sesuai dengan kebutuhan dari analisis persyaratan, semua *library* bersifat *open-source* sehingga bebas digunakan.

Tabel 6.1 Spesifikasi Perangkat Keras

Unit Komputasi	ASUS X455LF
CPU	Intel(R) Core(TM) i3-5005U 2.00GHz
Kapasitas RAM	4,00 GB
Kapasitas Penyimpanan	500 GB
Kartu Grafis	NVIDIA GEFORCE930M
Resolusi Layar	1366 x 768 pixels

Tabel 6.2 Spesifikasi Perangkat Lunak

Sistem Operasi	Windows 10 Enterprise Build 14393
Web Server	Apache 2.4.18
DBMS	MariaDB 10.1.10
Bahasa Pemrograman	PHP 7.0.3 (VC14 X86 32bit thread safe) + PEAR
Editor Kode Program	Sublime Text 3
Peramban	Google Chrome 67.0.3396.87 (Official Build) (64-bit) (cohort: Stable)
Bahasa Pemrograman Pendukung	HTML
	Javascript with jQuery
Library & Helper Pendukung	sendmail(PHP), Database (CI), Session (CI), Form Validation (CI), URL (CI), Form (CI), email (CI), Security (CI), jquery.validate

Tabel 6.2 Spesifikasi Perangkat Lunak (Lanjutan)

	(jquery), bootstrap-notify (jquery), datatables (jquery), bootstrap-table (jquery), bootstrap-select (jquery), bootstrap-datepicker (jquery) Bootstrap (CSS)
--	--

Tabel 6.3 Spesifikasi Lingkungan Instalasi

Kapasitas RAM	256 MB
Kapasitas Penyimpanan	50 MB
Versi PHP	PHP 7.0.3
Library Pendukung	sendmail(PHP)

Kapasitas RAM rekomendasi dari sistem adalah 256 MB berdasarkan *snapshot* dari RAM yang digunakan ketika masih dalam lingkungan pengembangan yaitu sekitar 11 MB jika menjalankan server http (apache) dan sekitar 2 MB ketika menjalankan server mysql, *snapshot* diambil ketika sistem diakses satu pengguna, dalam lingkungan instalasi nantinya sistem akan diakses oleh kurang lebih 5-10 pengguna (petugas /konselor) dan bisa lebih jika diakses juga oleh calon pasien, sehingga peneliti menyarankan kapasitas RAM rekomendasi sebesar 256 MB.

Sedangkan untuk penyimpanan adalah 50 MB, besar dari sistem ini sebelum di-*minify* adalah sekitar 24-25 MB sehingga *storage* 25 MB sisanya mampu untuk menampung foto profil atau menyimpan *record* di dalam *database*.

Lalu versi PHP 7.0.3 merupakan versi PHP yang digunakan pada CodeIgniter 3.1.8 dan karena peneliti mengembangkannya dalam lingkungan tersebut sehingga untuk dukungan semua fitur secara maksimal maka sangat disarankan agar lingkungan instalasi menggunakan versi PHP ini beserta *library* pendukung sendmail untuk mengirim email.

6.2 Implementasi Algoritme

Subbab ini berisi implementasi *pseudocode* dari bab perancangan menjadi baris kode program, hasil implementasi kode program dikelompokkan berdasarkan fitur yang dikembangkan yaitu masuk sistem, mengelola data pasien, mengelola data konseling, menambah data konseling, menambah data pribadi, ganti *password*, cetak formulir dan keluar sistem.

6.2.1 Masuk Sistem

Kode program pada Tabel 6.4 merupakan implementasi algoritme dari masuk sistem. Menggunakan *library security* dari CI untuk memastikan *input* dari pengguna bersih dari karakter membahayakan yang dapat menyebabkan *sql injection*.

Tabel 6.4 Kode Program Masuk Sistem

No	Kode Program
1	public function login(){
2	\$this->load->model('M_Pengguna');
3	\$username = \$this->security->xss_clean(
4	filter_input(INPUT_POST, 'username', FILTER_SANITIZE_STRING));
5	\$password = \$this->security->xss_clean(
6	filter_input(INPUT_POST, 'password', FILTER_SANITIZE_STRING));
7	\$user=\$this->M_Pengguna->get_pengguna(array('username'=>\$username),
8	array('username','password','tipe'));
9	if (\$user['username']) {
10	if (password_verify(\$password,\$user['password'])) {
11	\$data_session = array(
12	'username' => \$username,
13	'status' => "login",
14	'tipe' => \$user['tipe']);
15	\$this->session->set_userdata(\$data_session);
16	redirect('/');
17	}else{
18	\$this->session->set_flashdata('response', 'Password Salah!!');
19	redirect('masuk');
20	}else{
21	\$this->session->set_flashdata('response', 'Username tidak ada!!');
22	redirect('masuk');
23	}}

6.2.2 Mengelola data pasien

Kode program pada Tabel 6.5 merupakan implementasi algoritme dari mengelola data pasien. Menggunakan *library security* dari CI untuk memastikan *input* dari pengguna bersih dari karakter membahayakan yang dapat menyebabkan *sql injection*. Ada dua pilihan untuk mengubah data yaitu perubahan data atribut yang berkaitan informasi pribadi pasien dan perubahan data untuk verifikasi pasien tersebut, verifikasi dilakukan jika pasien telah datang langsung ke BNN untuk melakukan rehabilitasi.

Tabel 6.5 Kode Program Mengelola Data Pasien

No	Kode Program
1	public function ubah_pasien(\$id_pasien){
2	if (\$this->input->post('btn-ubah-pasien')== 'ubah') {
3	\$data=array('id'=>\$this->input->post('no_identitas'),
4	'jenis_id'=>\$this->input->post('tipe_identitas'),
5	'nama'=>\$this->input->post('fullname'),
6	'jenis_kelamin'=>\$this->input->post('jenis_kelamin'),
7	'no_telp'=>\$this->input->post('no_telp'),
8	'tempat_lahir'=>\$this->input->post('tempat_lahir'),
9	'tgl_lahir'=>date_format(date_create_from_format('d/m/Y',\$this->
10	input->post('tgl_lahir'),'Y/m/d'),
11	'alamat'=>\$this->input->post('alamat'),
12	'id_provinsi'=>\$this->input->post('provinsi'),
13	'id_agama'=>\$this->input->post('agama'),
14	'suku'=>\$this->input->post('suku'),
15	'goldar'=>\$this->input->post('goldar'),
16	'pend_terakhir'=>\$this->input->post('pend_terakhir'),
17	'pekerjaan'=>\$this->input->post('pekerjaan'),
18	'menikah'=>\$this->input->post('menikah'),
19	'nama_pasangan'=>\$this->input->post('nama_pasangan'),
20	'nama_ayah'=>\$this->input->post('nama_ayah'),

Tabel 6.5 Kode Program Mengelola Data Pasien (Lanjutan)

21	'nama_ibu'=>\$this->input->post('nama_ibu'),
22	'nomer_telp_keluarga'=>\$this->input->post('telp_keluarga'),
23	'alamat_keluarga'=>\$this->input->post('alamat_keluarga'),
24	'id_sumber_pasien'=>\$this->input->post('sumber_pasien'),
25	'id_sumber_biaya'=>\$this->input->post('sumber_biaya'),
26	'no_rekam_medis'=>\$this->input->post('no_rekam_medis'),
27	'mulai_pakai'=>date_format(date_create_from_format('d/m/Y',\$this->input->post('mulai_pakai')), 'Y/m/d'),
28	'terakhir_pakai'=>date_format(date_create_from_format('d/m/Y',\$this->input->post('terakhir_pakai')), 'Y/m/d'),
29	'cara_pakai'=>\$this->input->post('cara_pakai'),
30	'frekuensi'=>\$this->input->post('frekuensi'),
31	'pernah_rehabilitasi'=>\$this->input->post('rehabilitasi'),
32	'tempat_rehabilitasi'=>\$this->input->post('tempat_rehab'),
33	'status_rawat'=>\$this->input->post('rawat'),
34	'lama_rawat'=>\$this->input->post('lama_rawat'),
35	'email'=>\$this->input->post('email_pasien'));
36	\$narkoba = \$this->input->post('jenis_narkoba[]');
37	if (\$this->M_Pasien->perbarui_pasien(\$id_pasien,\$data,\$narkoba)) {
38	\$this->session->set_flashdata('response','Ubah data berhasil');
39	redirect('pasien');
40	}else{
41	\$this->session->set_flashdata('response','Ubah data gagal');
42	redirect('pasien');
43	}}}

6.2.3 Cetak formulir

Kode program pada Tabel 6.6 merupakan implementasi algoritme dari cetak formulir. Menggunakan *library security* dari CI untuk memastikan *input* dari pengguna bersih dari karakter membahayakan yang dapat menyebabkan *sql injection*. Pengkondisian menggunakan *switch-case* sehingga menghasilkan kode program yang lebih sedikit dan efektif.

Tabel 6.6 Kode Program Cetak Formulir

No	Kode Program
1	public function print(\$id_pasien,\$tipe_formulir){
2	\$id_pasien = \$this->security->xss_clean(\$id_pasien);
3	\$tipe_formulir = \$this->security->xss_clean(\$tipe_formulir);
4	\$data=\$this->M_Pasien->pasien(\$id_pasien);
5	\$sir=\$this->M_Formulir->pendaftaran(\$id_pasien);
6	if (substr(\$tipe_formulir,0,2)=='rk') {
7	\$konseling_ke=substr(\$tipe_formulir, 2);
8	\$konseling=\$this->M_Formulir->konseling(\$id_pasien,\$konseling_ke);
9	\$this->load->view('print/rk', array('data'=>\$konseling));
10	}
11	switch(\$tipe_formulir){
12	case 'dpn':
13	\$this->load->view('print/dpn', array('data'=>\$data));
14	break;
15	case 'sir':
16	\$this->load->view('print/sir', array('data'=>\$sir));
17	break;
18	case 'lpt':
19	\$this->load->view('print/lpt', array('data'=>\$data));
20	break;
21	case 'lpr':
22	\$this->load->view('print/lpr', array('data'=>\$data));
23	break;
24	case 'ak':
25	\$this->load->view('print/ak', array('data'=>\$data));
26	break;}}

6.2.4 Mengelola data konseling pasien

Kode program pada Tabel 6.7 merupakan implementasi algoritme dari Mengelola data konseling pasien. Menggunakan ajax dalam melakukan pertukaran data sehingga mengirimkan perintah dengan mencetak menggunakan *echo* lalu sisi klien akan menangkap perintah yang dikirimkan dari fungsi dalam *controller*.

Tabel 6.7 Kode Program Mengelola data konseling pasien

No	Kode Program
1	<code>public function ubah(){</code>
2	<code>\$id_konseling=\$this->input->post('id_konseling');</code>
3	<code>\$data=array('masalah_motivasi'=>\$this->input->post('masalah_motivasi'),</code>
	<code>'hal_yg_menghambat_penyelesaian'=>\$this->input->post('hal_yg_menghambat_penyelesaian'),</code>
	<code>'hal_yg_mendukung_penyelesaian'=>\$this->input->post('hal_yg_mendukung_penyelesaian'),</code>
	<code>'rencana_tindak_lanjut'=>\$this->input->post('rencana_tindak_lanjut'));</code>
4	<code>if (\$this->M_Konseling->perbarui(\$id_konseling,\$data)){</code>
5	<code>echo "ok";</code>
6	<code>}else{</code>
7	<code>echo "Data gagal diubah, mohon muat ulang halaman lalu ulangi</code>
8	<code>lagi";</code>
9	<code>}}</code>

6.2.5 Menambah data konseling pasien

Kode program pada Tabel 6.8 merupakan implementasi algoritme dari Menambah data konseling pasien. Menggunakan *library security* untuk melakukan pembersihan dari masukkan pengguna.

Tabel 6.8 Kode Program Menambah data konseling pasien

No	Kode Program
1	<code>public function konseling_detail(\$id_pasien){</code>
2	<code>if (\$this->input->post('btn-tambah-konseling')) {</code>
3	<code>\$data=array('id_pasien'=>\$id_pasien,</code>
4	<code>'id_konselor'=>\$id_konselor,</code>
5	<code>'konseling_ke'=>(\$banyak_konseling+1),</code>
6	<code>'masalah_motivasi'=>\$this->input->post('masalah_motivasi'),</code>
7	<code>'hal_yg_menghambat_penyelesaian'=>\$this->input->post('hal_yg_menghambat_penyelesaian'),</code>
8	<code>'hal_yg_mendukung_penyelesaian'=>\$this->input->post('hal_yg_mendukung_penyelesaian'),</code>
9	<code>'rencana_tindak_lanjut'=>\$this->input->post('rencana_tindak_lanjut'));</code>
10	<code>if (\$this->M_Konseling->tambah(\$data)) {</code>
11	<code>\$this->session->set_flashdata('response','Data konseling berhasil</code>
12	<code>ditambahkan');</code>
13	<code>redirect("konseling/\$id_pasien",'refresh');</code>
14	<code>}else{</code>
15	<code>\$this->session->set_flashdata('response','Data konseling gagal</code>
	<code>ditambahkan');</code>
16	<code>redirect("konseling/\$id_pasien",'refresh');</code>
17	<code>}}</code>

6.2.6 Menambahkan data pribadi

Kode program pada Tabel 6.9 merupakan implementasi algoritme dari Menambahkan data pribadi. Dalam implementasi menggunakan *recaptcha* untuk mencegah *request* yang tidak sah yang disebabkan oleh *macro* (bot) yang dapat membebani sistem lewat celah ini.

Tabel 6.9 Kode Program Menambahkan data pribadi

No	Kode Program
1	<code>public function hasil(){</code>
2	<code>\$captcha=\$this->input->post('g-recaptcha-response');</code>
3	<code>\$secret='6LdLYloUAAAAADtdbsQ3Oy6PrsHPjiEg5Uuc1nP';</code>
4	<code>\$response = json_decode(file_get_contents</code> <code>("https://www.google.com/recaptcha/api/</code> <code>siteverify?secret=".\$secret."&response=".\$captcha."&remoteip=</code> <code>".\$_SERVER["REMOTE_ADDR"]), true);</code>
5	<code>\$data=array('id'=>\$this->input->post('no_identitas'),</code>
6	<code>'jenis_id'=>\$this->input->post('tipe_identitas'),</code>
7	<code>'nama'=>\$this->input->post('fullname'),</code>
8	<code>'jenis_kelamin'=>\$this->input->post('jenis_kelamin'),</code>
9	<code>'no_telp'=>\$this->input->post('no_telp'),</code>
10	<code>'tempat_lahir'=>\$this->input->post('tempat_lahir'),</code>
11	<code>'tgl_lahir'=>date_format(date_create_from_format</code> <code>('d/m/Y', \$this->input->post('tgl_lahir')), 'Y/m/d'),</code>
12	<code>'alamat'=>\$this->input->post('alamat'),</code>
13	<code>'id_provinsi'=>\$this->input->post('provinsi'),</code>
14	<code>'id_agama'=>\$this->input->post('agama'),</code>
15	<code>'suku'=>\$this->input->post('suku'),</code>
16	<code>'goldar'=>\$this->input->post('goldar'),</code>
17	<code>'pend_terakhir'=>\$this->input->post('pend_terakhir'),</code>
18	<code>'pekerjaan'=>\$this->input->post('pekerjaan'),</code>
19	<code>'menikah'=>\$this->input->post('menikah'),</code>
20	<code>'nama_pasangan'=>\$this->input->post('nama_pasangan'),</code>
21	<code>'nama_ayah'=>\$this->input->post('nama_ayah'),</code>
22	<code>'nama_ibu'=>\$this->input->post('nama_ibu'),</code>
23	<code>'nomer_telp_keluarga'=>\$this->input->post('telp_keluarga'),</code>
24	<code>'alamat_keluarga'=>\$this->input->post('alamat_keluarga'),</code>
25	<code>'id_sumber_pasien'=>\$this->input->post('sumber_pasien'),</code>
26	<code>'id_sumber_biaya'=>\$this->input->post('sumber_biaya'),</code>
27	<code>'no_rekam_medis'=>\$this->input->post('no_rekam_medis'),</code>
28	<code>'mulai_pakai'=>date_format(date_create_from_format</code> <code>('d/m/Y', \$this->input->post('mulai_pakai')), 'Y/m/d'),</code>
29	<code>'terakhir_pakai'=>date_format(date_create_from_format</code> <code>('d/m/Y', \$this->input->post('terakhir_pakai')), 'Y/m/d'),</code>
30	<code>'cara_pakai'=>\$this->input->post('cara_pakai'),</code>
31	<code>'frekuensi'=>\$this->input->post('frekuensi'),</code>
32	<code>'pernah_rehabilitasi'=>\$this->input->post('rehabilitasi'),</code>
33	<code>'tempat_rehabilitasi'=>\$this->input->post('tempat_rehab'),</code>
34	<code>'status_rawat'=>\$this->input->post('rawat'),</code>
35	<code>'lama_rawat'=>\$this->input->post('lama_rawat'),</code>
36	<code>'email'=>\$this->input->post('email');</code>
37	<code>\$narkoba = \$this->input->post('jenis_narkoba[]');</code>
38	<code>if (\$response['success']) {</code>
39	<code>if(\$this->M_Pasien->tambah_pasien(\$data,\$narkoba)){</code>
40	<code>if(\$this-> kirim_email(\$data['email'],</code>
41	<code>"Konfirmasi Pendaftaran Rehabilitasi - BNN Kota Malang",</code> <code>"Terimakasih ".\$data['nama']." telah mendaftarkan diri,</code> <code>dapat langsung datang dengan membawa kartu identitas yang</code> <code>anda daftarkan untuk mem-verifikasi")){</code>
42	<code>\$this->session->set_flashdata('konfirmasi', 'Data telah terekam</code> <code>& Kirim email berhasil');</code>

Tabel 6.9 Kode Program Menambahkan data pribadi (Lanjutan)

43	<code>\$this->load->view('daftar/halaman-sukses',array('userdata'=>array('nama'=>\$data['nama'],</code>
44	<code>'email'=>\$data['email'])));</code>
45	<code>}else{</code>
	<code>\$this->session->set_flashdata('konfirmasi',</code>
46	<code>'Data telah terekam & Kirim email gagal, harap cek alamat email</code>
	<code>anda');</code>
	<code>\$this->load->view('daftar/halaman-sukses',array('userdata'=>array('nama'=>\$data['nama'],</code>
47	<code>'email'=>\$data['email'])));</code>
48	<code>}else{</code>
49	<code>\$this->session->set_flashdata('konfirmasi', 'Data belum masuk');</code>
50	<code>\$this->load->view('daftar/halaman-sukses');</code>
51	<code>}else{</code>
52	<code>\$this->session->set_flashdata('konfirmasi', 'Belum Captcha/Captcha</code>
	<code>tidak valid');</code>
53	<code>\$this->load->view('daftar/halaman-sukses');</code>
54	<code>}}</code>
55	<code>private function kirim_email(\$to,\$subject,\$message){</code>
56	<code>\$status;</code>
57	<code>\$this->load->library('email');</code>
58	<code>\$config=array(</code>
59	<code>'protocol'=>'smtp',</code>
60	<code>'smtp_host'=>'ssl://smtp.gmail.com',</code>
61	<code>'smtp_port'=>'465',</code>
62	<code>'smtp_user'=>'handosub@gmail.com',</code>
63	<code>'smtp_pass'=>'*****',</code>
64	<code>'mailtype'=>'html',</code>
65	<code>'charset'=>'iso-8859-1',</code>
66	<code>'wordwrap'=>TRUE);</code>
67	<code>\$this->email->initialize(\$config);</code>
68	<code>\$this->email->from('no-reply@bnnkotamalang.com','ADMIN');</code>
69	<code>\$this->email->to(\$to);</code>
70	<code>\$this->email->subject(\$subject);</code>
71	<code>\$this->email->message(\$message);</code>
72	<code>if(\$result=\$this->email->send()){</code>
73	<code>\$status=true;</code>
74	<code>}else{</code>
75	<code>\$status=false;</code>
	<code>return \$status;</code>

6.2.7 Ganti password

Kode program pada Tabel 6.10 merupakan implementasi algoritme dari Ganti *password*. Menggunakan ajax dalam melakukan pertukaran data sehingga mengirimkan perintah dengan mencetak menggunakan *echo* lalu sisi klien akan menangkap perintah yang dikirimkan dari fungsi dalam *controller*.

Tabel 6.10 Kode Program Ganti password

No	Kode Program
1	<code>public function ganti_password(){</code>
2	<code>\$this->load->model('M_Pengguna');</code>
3	<code>\$email=\$this->input->post('email');</code>
4	<code>\$old_password=\$this->input->post('old_password');</code>
5	<code>\$new_password=\$this->input->post('new_password');</code>
6	<code>\$passconf=\$this->input->post('passconf');</code>
7	<code>\$data=\$this->M_Pengguna->get_pengguna(array('email'=>\$email));</code>
8	<code>if (\$data) {</code>
9	<code>if (password_verify(\$old_password,\$data['password'])) {</code>
10	<code>if (strcmp(\$new_password,\$passconf)==0) {</code>
11	<code>if (!(strcmp(\$new_password,\$old_password)==0)) {</code>

Tabel 6.10 Kode Program Ganti password (Lanjutan)

12	<code>\$this->M_Pengguna-</code>
	<code>>ganti_password(\$email,password_hash(\$new_password,PASSWORD_DEFAULT));</code>
13	<code>if (\$this->db->affected_rows() > 0) {</code>
14	<code>echo "ok";</code>
15	<code>}else{</code>
16	<code>echo "Terjadi kesalahan sistem, coba ulangi lagi";</code>
17	<code>}else{</code>
18	<code>echo "Password baru tidak boleh sama dengan password lama";</code>
19	<code>}else{</code>
20	<code>echo "Password baru dan konfirmasi tidak sama";</code>
21	<code>}else{</code>
22	<code>echo "password lama salah";}</code>
23	<code>}else{</code>
24	<code>echo "email salah";}}</code>

6.2.8 Keluar Sistem

Kode program pada Tabel 6.11 merupakan implementasi algoritme dari Keluar Sistem, sesuai seperti perancangan yaitu menghapus *session* dari pengguna yang bersangkutan lalu mengalihkan ke *controller* masuk.

Tabel 6.11 Kode Program Keluar Sistem

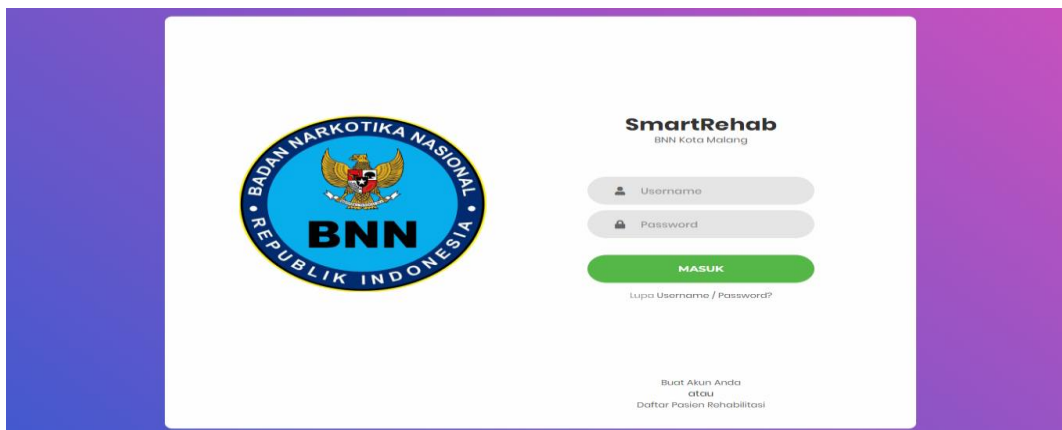
No	Kode Program
1	<code>public function logout(){</code>
2	<code> \$this->session->sess_destroy();</code>
3	<code> redirect('masuk');</code>
4	<code>}</code>

6.3 Implementasi Antarmuka pengguna

Pada subbab ini berisi tampilan-tampilan antarmuka pengguna hasil pengembangan sistem berdasarkan sketsa antarmuka pengguna dari tahap perancangan (bab sebelumnya). Peneliti menampilkan semua antarmuka pengguna sesuai hasil sketsa antarmuka pengguna pada tahap perancangan.

6.3.1 Antarmuka Masuk sistem

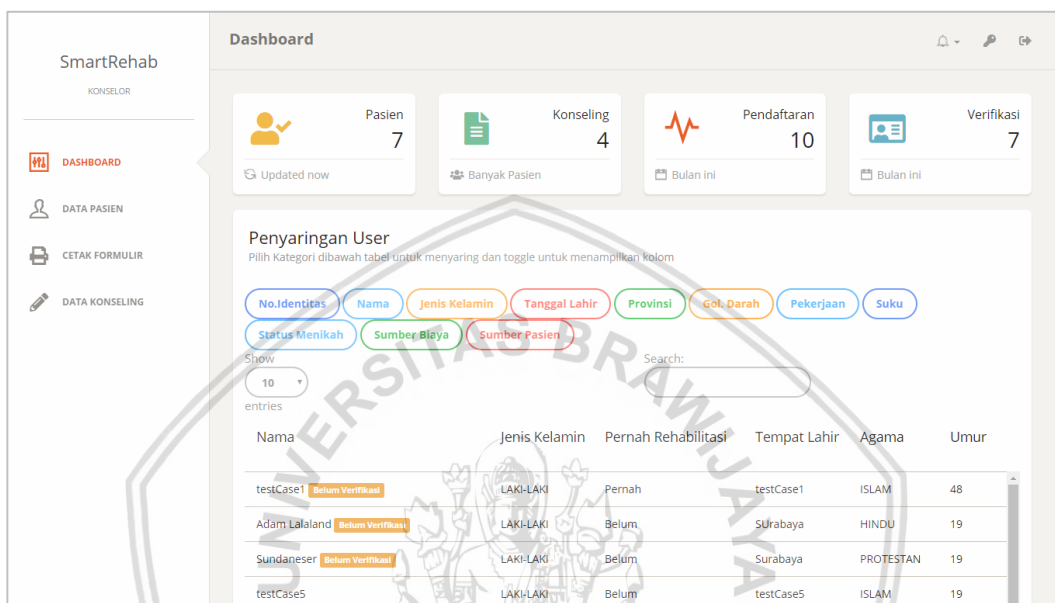
Pada gambar di bawah ini merupakan antarmuka untuk masuk ke sistem, logo di sebelah kiri dan *form* untuk masukkan *username* dan *password* beserta tombol untuk masuk, pada antarmuka ini disediakan jarak antar masukkan dan tombol untuk memberikan informasi terkait kegagalan pada proses masuk sistem.



Gambar 6.1 Antarmuka Masuk Sistem

6.3.2 Antarmuka *Dashboard*

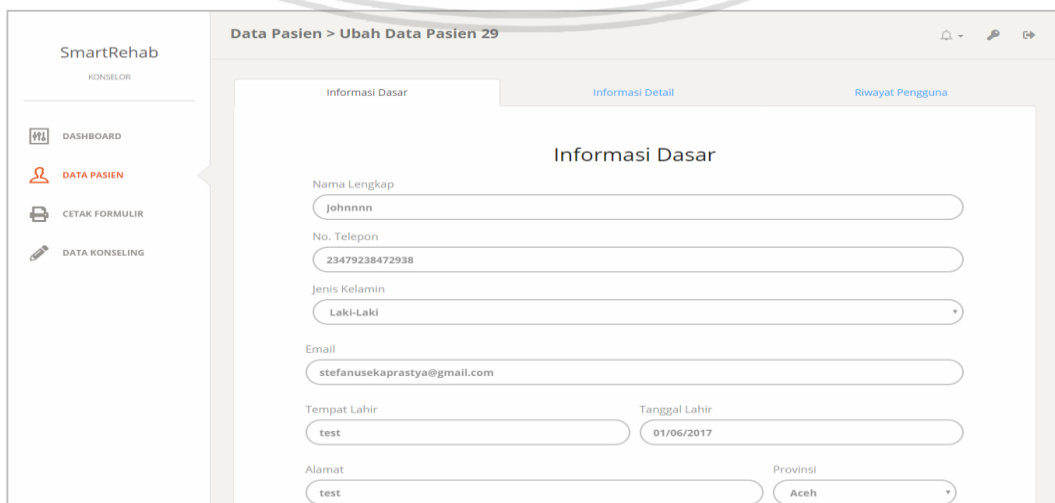
Antarmuka *dashboard* merupakan tampilan jika berhasil masuk ke dalam sistem, berisi beberapa “*card*” yang berisi informasi terkini terkait sistem informasi data pasien, di sebelah kiri tersedia *sidebar* untuk memudahkan navigasi antar menu dalam sistem informasi di pojok kanan atas terdapat dua menu tambahan yaitu notifikasi dan pengaturan untuk mengganti *password* atau keluar sistem.



Gambar 6.2 Antarmuka *Dashboard*

6.3.3 Antarmuka Mengelola data pasien

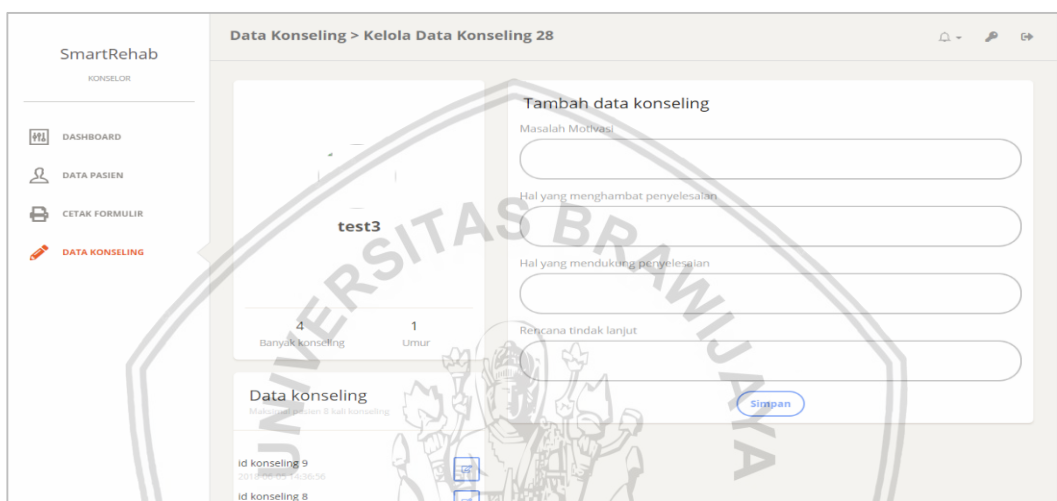
Gambar di bawah ini merupakan antarmuka untuk mengelola data pasien, penulis menggunakan tab untuk membagi *input* menjadi tiga bagian untuk mempermudah pengubahan/penulisan data dan meningkatkan ketelitian dalam mengubah data karena tidak secara langsung disodorkan banyak *input* dalam satu halaman.



Gambar 6.3 Antarmuka Mengelola data pasien

6.3.4 Antarmuka Mengelola data konseling pasien

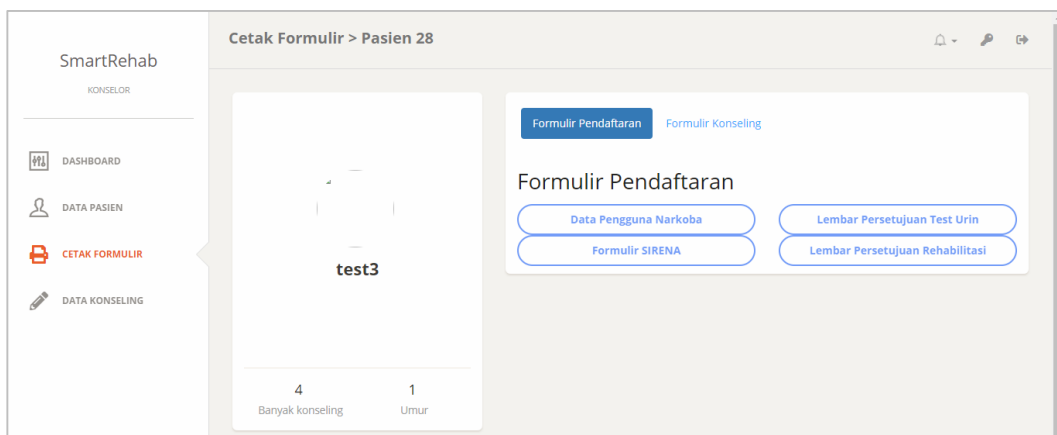
Antarmuka ini berbeda dengan tampilan untuk mengelola data pasien, karena dalam rancangan tampilan ini terdapat *section* untuk menampilkan *profile picture* (jika pasien mengunggahnya), umur dan banyak konseling, lalu di bawahnya menampilkan data konseling terdahulu dari pasien, pengguna bisa mengubah atau menghapusnya, di sebelah kanan terdapat *textarea* untuk memasukkan data konseling dan langsung menyimpannya, *textarea* di sebelah kiri akan muncul jika jumlah konseling dari pasien tersebut di bawah delapan, karena sesuai dengan aturan yang berlaku satu pasien hanya bisa konseling sebanyak delapan kali.



Gambar 6.4 Antarmuka Mengelola data konseling pasien

6.3.5 Antarmuka Cetak formulir

Antarmuka untuk cetak formulir hampir sama seperti data konseling pasien, karena di sebelah kiri terdapat "card" yang berisi *profile picture*, nama, banyak konseling dan umur, tetapi di sebelah kanan ada dua tab yang berisi beberapa tombol untuk mencetak formulir, untuk tipe akun petugas hanya bisa mencetak pada formulir yang tersedia pada tab formulir pendaftaran, lalu untuk tipe akun konselor dapat mencetak semua formulir.



Gambar 6.5 Antarmuka Cetak Formulir

6.3.6 Antarmuka Ganti *password*

Antarmuka untuk ganti *password* adalah dengan menampilkan “modal” yang berisi masukkan sebagai berikut (Gambar 6.6), tampilan “modal” dipilih karena lebih ringkas dan cepat (tidak perlu pindah halaman) untuk mengganti *password* dapat langsung meng-klik menu tambahan di sebelah kanan atas.

Gambar 6.6 Antarmuka Ganti Password

6.3.7 Antarmuka Menambahkan data pribadi

Antarmuka untuk menambahkan data pribadi terdiri dari banyak masukkan, sehingga penulis menggunakan “wizard” yang terdiri dari tiga tab yang mempunyai pembagian seperti pada fitur mengelola data pasien, lalu disediakan fitur untuk mem-validasi masukkan dari *user*, sehingga *form* lebih intuitif dan memastikan masukkan dari pasien benar dan valid.

Gambar 6.7 Antarmuka Menambahkan data pribadi

BAB 7 PENGUJIAN

Bab ini membahas hasil pengujian yang dilakukan sesuai dengan perencanaan kasus uji dari bab 5, pengujian yang dilakukan antara lain adalah pengujian skenario apakah pengguna berhasil melakukan skenario sesuai hasil yang diharapkan, lalu pengujian *usability* untuk menguji apakah sistem yang dikembangkan mudah untuk digunakan dan pengujian terakhir adalah pengujian kompatibilitas browser. Pengujian kompatibilitas browser dan *usability* sesuai dengan hasil wawancara validasi masalah bagaimana *stakeholder* membutuhkan sistem yang bisa dibuka menggunakan platform apa saja dan mudah dipelajari dan digunakan.

7.1 Pengujian Skenario

Subbab ini berisi bagaimana peneliti melakukan pengujian skenario, pertama adalah merancang kasus uji (bab 5) lalu dilanjutkan dengan mengeksekusi pengujian, pengujian dilakukan oleh peneliti lalu dilaporkan kepada *stakeholder*, sesuai dengan metodologi penelitian (bab 3), kelebihan pengujian menggunakan metode ini adalah pengujian tidak terbatas menilai apakah satu fungsionalitas dari sistem berjalan sesuai yang direncanakan (*valid*) tetapi bisa menguji lebih dari satu fungsionalitas dan bisa saja hasil pengujian skenario berhasil tetapi ditemukan *bugs*, hal-hal yang tidak dikehendaki *stakeholder* atau hal yang ingin ditambahkan, dengan menggunakan pengujian skenario diharapkan segala *bugs* yang tidak ditemukan pada pengujian validasi dapat ditemukan.

7.1.1 Hasil Pengujian Skenario

Hasil pengujian menggunakan metode ini terdiri dari 3 hasil untuk setiap kasus uji yaitu hasil pengujian apakah skenario berhasil dijalankan, skenario dikatakan berhasil hanya jika memenuhi kondisi yang diharapkan dari kasus uji yang dirancang, lalu yang kedua adalah apakah ada *bugs* yang ditemukan, karena mungkin saja skenario berhasil dijalankan tetapi masih menyisakan *bugs*, contoh alur dari sistem tidak sesuai dengan prosedur dari organisasi dan yang ketiga adalah tambahan fitur yang dibutuhkan oleh *stakeholder* yang mungkin baru muncul ketika pengujian ini dijalankan. Tujuan dari hasil pengujian ini adalah tidak hanya mencari kesalahan (*bugs*) saja tetapi untuk beberapa skenario pengujian untuk membuktikan pengembangan sistem sesuai seperti yang direncanakan yaitu dapat mempercepat aktivitas dan dengan hasil ini dapat menemukan kesepakatan atau jalan keluar dari *bugs* tersebut.

7.1.1.1 Skenario menambah data pribadi

Tabel di bawah ini merupakan hasil pengujian dari skenario menambah data pribadi, di kasus uji ini peneliti mencoba skenario penggunaan ahli yang merupakan pertanyaan dari para *stakeholder* ketika peneliti mencoba menyarankan pengembangan sistem ini yaitu bagaimana jika calon pasien yang ingin mendaftar lebih dari satu pada waktu yang bersamaan, pengujian ini

membuktikan bagaimana sistem mampu menangani permintaan pendaftaran lebih dari satu pasien, karena sistem ini mampu diakses dari berbagai platform termasuk *mobile*, karena unit *pc* yang tersedia di kantor BNN untuk saat ini tidak banyak jadi hal ini dikhawatirkan oleh *stakeholder* karena jika *mengantri* akan memakan lebih banyak waktu dibanding dengan cara yang konvensional (mengisi formulir), tetapi hasil pengujian membuktikan bahwa skenario berhasil dan waktu yang dihabiskan jauh lebih cepat dari batas waktu yang diberikan tetapi masih terdapat saran penambahan fitur oleh *stakeholder* untuk memasukkan foto ke dalam formulir dan fitur untuk mengelola tanda tangan atasan, fitur ini tidak muncul ketika analisis persyaratan tetapi ketika pengujian ini dijalankan karena dapat disadari bahwa para atasan tidak selalu hadir di kantor dan bisa jadi atasan diganti, jadi fitur ini dibutuhkan oleh *stakeholder*.

Tabel 7.1 Hasil Pengujian Skenario Menambah data pribadi

Pendaftaran Pasien Baru	ST-01
Tujuan Pengujian	Menguji apakah sistem dapat mendaftarkan pasien baru
Skenario Pengujian	<ol style="list-style-type: none"> 1. Calon pasien mengisi <i>form</i> dan mengumpulkan menggunakan SmartRehab 2. Pasien yang mendaftar sebanyak 3 orang bersamaan, 2 orang menggunakan platform <i>mobile</i>, satu orang menggunakan desktop browser 3. Petugas melakukan verifikasi lalu mencetak formulir pendaftaran
Hasil yang diharapkan	Formulir tercetak sesuai data dari pasien dan data tersimpan di dalam <i>database</i>
Batas Waktu	5 jam 3 menit 30 detik
Hasil Pengujian	Skenario berhasil
Waktu yang dihabiskan	31 menit 12 detik
<i>Bugs</i> yang ditemukan	Tidak ditemukan <i>bugs</i>
Saran penambahan fitur	<ul style="list-style-type: none"> • Memasukkan foto ke dalam formulir • Membuat fitur untuk mengelola tanda tangan atasan

7.1.1.2 Skenario pencarian berdasarkan kategori

Tabel di bawah ini merupakan hasil pengujian dari skenario pencarian berdasarkan kategori, dengan kasus uji peneliti mencoba membuktikan bahwa sistem ini dapat menghasilkan manfaat daripada cara yang konvensional, manfaat ini merupakan kebutuhan utama dari *stakeholder* dan dengan pengujian skenario dapat dipastikan sistem ini dapat menghasilkan manfaat ini dengan waktu yang lebih cepat, tetapi munculnya saran penambahan fitur yang mungkin tidak muncul ketika analisis persyaratan yaitu pemuatan data secara dinamis, mengingat sistem ini akan dipakai untuk jangka waktu yang panjang sehingga data akan terus bertumbuh, jika data dimuat secara statis mungkin untuk banyak data sebanyak data contoh yang digunakan untuk penelitian ini tidak terjadi proses yang lama, tetapi jika data mencapai ratusan atau hingga ribuan maka kecepatan dalam memuat data tidak akan secepat ketika menggunakan data contoh.

Tabel 7.2 Hasil Pengujian Skenario Pencarian berdasarkan kategori

Pencarian berdasarkan kategori	ST-02
Tujuan Pengujian	Menguji apakah sistem dapat menampilkan hasil pencarian berdasarkan kategori
Skenario Pengujian	<ol style="list-style-type: none"> 1. Petugas membuat <i>query</i> pencarian berdasarkan kategori 2. Petugas mencari data berdasarkan kategori 3. Sistem menampilkan hasil pencarian
Hasil yang diharapkan	Sistem menampilkan hasil pencarian
Batas Waktu	21 detik
Hasil Pengujian	Skenario berhasil
Waktu yang dihabiskan	19 detik
<i>Bugs</i> yang ditemukan	Tidak ditemukan <i>bugs</i>
Saran penambahan fitur	Proses pemuatan data secara dinamis

7.1.1.3 Skenario menambah data konseling

Tabel di bawah ini merupakan hasil pengujian skenario menambah data konseling, dengan kasus uji ini membuktikan bahwa skenario berhasil tanpa menemukan *bugs*, tetapi terdapat saran untuk menambahkan fitur untuk mengelola tandatangan konselor, fitur ini sama seperti pada kasus uji skenario pertama tetapi subjeknya yang berbeda yaitu tandatangan dari konselor.

Tabel 7.3 Hasil Pengujian Skenario menambah data konseling

Menambah data konseling	ST-03
Tujuan Pengujian	Menguji apakah sistem dapat menyimpan data konseling
Skenario Pengujian	<ol style="list-style-type: none"> 1. Konselor melakukan konseling terhadap pasien 2. Konselor memasukkan data ke dalam <i>form</i> lalu menyimpannya 3. Konselor mencetak <i>form</i>
Hasil yang diharapkan	Formulir tercetak sesuai data konseling dari pasien dan data konseling tersimpan di dalam <i>database</i>
Hasil Pengujian	Skenario berhasil
<i>Bugs</i> yang ditemukan	Tidak ditemukan <i>bugs</i>
Saran penambahan fitur	Membuat fitur untuk mengelola tanda tangan konselor

7.1.2 Kesimpulan Pengujian Skenario

Dari tiga kasus uji skenario yang diuji didapatkan beberapa kasus uji yang merupakan kasus uji untuk penggunaan ahli (menambah data pribadi), lalu kasus uji untuk menguji manfaat utama (pencarian berdasarkan kategori), beberapa *insight* seperti saran penambahan fitur muncul melalui pengujian ini, pengujian skenario memiliki kelebihan seperti diatas jika digunakan pada kasus yang tepat. Hasil pengujian membuktikan tiga skenario dapat dijalankan berhasil pada sistem informasi ini.

7.2 Pengujian *Usability*

Pengujian *usability* dieksekusi dengan enam orang yang menjadi responden, pengujian ini dilakukan guna mengetahui tingkat kegunaan, kepuasan, kemudahan pengguna untuk menggunakan dan mempelajari sistem ini, seperti yang telah dirancang pada bab 5, keenam responden adalah staf rehabilitasi dan bagian umum yang nantinya akan menjadi pengguna dari sistem ini.

Sebelum melakukan pengujian *usability* peneliti memberikan kesempatan kepada *stakeholder* untuk mengamati ketika pengujian skenario lalu diperkenankan untuk menguji skenario, sehingga *stakeholder* mengetahui bagaimana alur, cara kerja dan antarmuka pengguna dari sistem.

7.2.1 Hasil Pengujian *Usability*

Hasil pengujian dari pengujian *usability* menggunakan skala *likert* 5 (interval 20), sehingga untuk menghitung nilai indeks pertanyaan menggunakan skala seperti pada tabel di bawah ini. Sistem akan dikatakan lolos dari pengujian ini jika persentase dari setiap kriteria diatas 60%, hasil perhitungan dari pengujian ini dapat dilihat pada Tabel 7.4.

Tabel 7.4 Interpretasi skor *Likert*

Skor Likert	Interpretasi skor dengan interval = 20	Pilihan
1	0% - 19,99%	Sangat tidak setuju
2	20% - 39,99%	Tidak setuju
3	40% - 59,99%	Netral
4	60% - 79,99%	Setuju
5	80% - 100%	Sangat setuju

Indeks persentase didapatkan dari Interpretasi skor *Likert* yang diterapkan pada pengujian *usability* dari 30 paket pertanyaan yang diberikan pada Tabel 5.13 yaitu menggunakan 5 skor. Indeks skor *likert* pada tabel di bawah ini didapatkan dengan memasukkan hasil rekap yang dihitung menggunakan Persamaan 2.1. Berikut ini merupakan contoh perhitungan indeks persentase skor *Likert* untuk pertanyaan nomor 1 didapatkan indeks didapatkan nilai sebesar 93%.

Tabel 7.5 Hasil Pengujian *Usability*

NO	PERNYATAAN	JAWABAN					Indeks
		STS	TS	N	S	SS	
		1	2	3	4	5	
Usefulness							
1	Sistem ini membantu saya dalam mengelola data rehabilitasi menjadi lebih efektif				2	4	93%
2	Sistem ini membantu saya dalam mengelola data rehabilitasi menjadi lebih produktif				4	2	87%
3	Sistem ini sangat berguna bagi saya				1	5	97%
4	Sistem ini memberikan saya penguasaan lebih dalam mengelola data rehabilitasi				5	1	83%
5	Sistem ini mempermudah saya mengelola data rehabilitasi				4	2	87%
6	Sistem ini membuat saya menghemat waktu saat mengelola data rehabilitasi			1	2	3	87%

Tabel 7.5 Hasil Pengujian *Usability* (Lanjutan)

7	Sistem ini memenuhi kebutuhan saya dalam mengelola data rehabilitasi			1	3	2	83%
8	Sistem ini sesuai dengan apa yang saya harapkan			1	5		77%
Rata-rata <i>Usefulness</i>		87%					
Easy to Learn							
9	Saya dapat mempelajari menggunakan Sistem ini dengan cepat			3	3		70%
10	Saya mudah mengingat bagaimana menggunakan Sistem ini			2	4		73%
11	Saya dengan mudah menggunakan Sistem ini				6		80%
12	Saya dengan cepat menguasai Sistem ini			3	3		70%
Rata-rata <i>Easy to Learn</i>		73%					
Easy to Use							
13	Sistem ini mudah digunakan				5	1	83%
14	Sistem ini dapat digunakan dengan sederhana				5	1	83%
15	Sistem ini dapat digunakan oleh semua kalangan masyarakat			4	1	1	70%
16	Sistem ini memerlukan langkah-langkah sederhana untuk dapat mencapai apa yang saya inginkan			1	4	1	80%
17	Sistem ini dapat digunakan secara fleksibel				5	1	83%
18	Tidak membutuhkan banyak usaha untuk menggunakan Sistem ini			1	4	1	80%
19	Saya dapat menggunakan Sistem ini tanpa instruksi tertulis			3	2	1	73%
20	Saya tidak menemukan inkonsistensi (ketidakesuaian) selama menggunakan Sistem ini			2	4		73%

Tabel 7.5 Hasil Pengujian *Usability* (Lanjutan)

21	Baik pengguna reguler maupun yang bukan reguler, akan menyukai Sistem ini			2	4		73%
22	Saya dapat dengan cepat dan mudah memperbaiki kesalahan saya dalam menggunakan Sistem			1	3	2	83%
23	Saya dapat menggunakan Sistem ini dengan praktis saat membutuhkannya				5	1	83%
Rata-rata <i>Easy to Use</i>			79%				
<i>Satisfaction</i>							
24	Saya puas dengan Sistem ini				5	1	83%
25	Saya akan merekomendasikan Sistem ini kepada orang lain				3	3	90%
26	Sistem ini menyenangkan untuk digunakan			1	3	2	83%
27	Sistem ini bekerja seperti yang saya inginkan				5	1	83%
28	Sistem ini sangat bagus				4	2	87%
29	Saya ingin memiliki Sistem ini			2	3	1	77%
30	Sistem ini nyaman untuk digunakan			1	4	1	80%
Rata-rata <i>Satisfaction</i>			83%				

7.2.2 Kesimpulan Pengujian *Usability*

Tabel 7.6 Analisis Pengujian *Usability*

Aspek Penilaian	Rata-rata Persentase (%)	Status
<i>Usefulness</i>	87	Sangat setuju
<i>Easy to learn</i>	73	Setuju
<i>Easy to use</i>	79	Setuju
<i>Satisfaction</i>	83	Sangat setuju
Rata-rata	81	Sangat Setuju

Hasil pengujian *usability* menunjukkan bahwa secara keseluruhan sistem memenuhi indeks *usability* dan sangat setuju bahwa sistem sangat berguna, untuk kriteria sebesar 87% yang membuktikan bahwa sistem sangat berguna, untuk kriteria *Easy to learn* dan *Easy to use* yang membuktikan bahwa sistem mudah

untuk digunakan dan dipelajari karena diatas indeks 60% sedangkan rata-rata *stakeholder* puas dengan sistem ini.

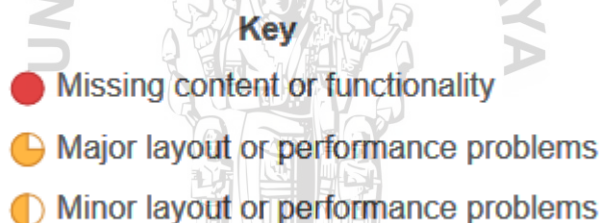
7.3 Pengujian Kompatibilitas

Pengujian ini dilakukan demi mengetahui seberapa besar tingkat kompatibilitas sistem yang dikembangkan dengan platform web pada beberapa platform sesuai dengan kasus uji yang telah dirancang pada bab 5, pemilihan metode ini bukan tanpa alasan tetapi demi membuktikan sistem yang dikembangkan sesuai dengan kebutuhan non-fungsional dari *stakeholder* yang didapatkan pada tahap analisis persyaratan.

7.3.1 Hasil Pengujian Kompatibilitas

Browser Version	IE			Edge	Firefox	Safari		Opera	Chrome	iOS			Android		BlackBerry	
	9	10	11	16	60	≤ 10	11	51	66	≤ 9	10	11	≤ 3	4*	≤ 7.1	10.0
Critical Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Major Issues	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Minor Issues	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

Gambar 7.1 Hasil Pengujian Kompatibilitas



Gambar 7.2 Skor Kompatibilitas

Pengujian dilakukan oleh tester dimana adalah peneliti sendiri menggunakan *tools* SortSite, hasil pengujian membuktikan bahwa mayoritas tidak ada masalah yang bersifat *critical* pada tiap platform, tetapi masih terdapat masalah mayor untuk platform internet explorer dan blackberry, dan masalah bersifat minor untuk browser firefox 60. Secara keseluruhan tidak ada ikon grafik lingkaran merah yang berarti pada sistem ini tidak ada konten yang hilang atau fungsionalitas yang tidak berjalan.

7.3.2 Kesimpulan Pengujian Kompatibilitas Browser

Hasil pengujian menunjukkan bahwa agar pengguna mendapatkan pengalaman pengguna yang penuh dan maksimal seperti yang diharapkan adalah dengan menggunakan browser versi terbaru seperti safari 11 atau opera 51 dan chrome 66, menggunakan platform lainnya dimungkinkan tetapi tidak mendapatkan pengalaman pengguna yang sepenuhnya. Hasil pengujian ini membuktikan sistem kurang memiliki kompatibilitas terhadap browser atau platform yang lawas.

BAB 8 PENUTUP

8.1 Kesimpulan

1. Dalam tahap analisis persyaratan peneliti mengumpulkan data dengan menggunakan metode *in-depth interview* untuk mengumpulkan masalah, dimana dari hasil analisis persyaratan didapatkan 5 fitur, 10 persyaratan fungsional dan 2 persyaratan non fungsional yang lalu menjadi 8 *use case*. Hasil analisis persyaratan juga memuat spesifikasi *use case* yang dimodelkan aktivitasnya menjadi diagram aktivitas yang berisi urutan aktivitas untuk menjalankan *use case*.
2. Penggunaan pendekatan berorientasi objek menjadi jembatan untuk menggabungkan proses serta data yang mana adalah atribut dari objek itu sendiri, perancangan dimulai dengan memodelkan interaksi antar objek dengan menggunakan *sequence diagram*, dari *sequence diagram* perancangan dilanjutkan ke visualisasi diagram kelas yang merupakan blok yang berisi atribut beserta *behaviour* dari objek nantinya, lalu dari diagram kelas analisis menjadi rancangan untuk tabel, dimana terdapat 10 kelas analisis, 9 tabel, 4 model dan 6 *controller* dimana memudahkan tahap perancangan untuk dilanjutkan ke tahap implementasi dalam pengembangan sistem informasi ini, ditambah dengan penggunaan pola mvc dengan *framework* CodeIgniter di dalam penerapan sistem ini membantu peneliti untuk memisahkan baris kode yang mungkin akan sangat sulit untuk di *maintain* jika tidak menggunakan pola mvc (*spaghetti code*).
3. Penggunaan metode strategi menggunakan *iterative* dan *incremental* menjadi kunci utama pengembangan sistem ini, dengan menggunakan dekomposisi fungsional yang menghasilkan 8 inkremen. Walau hasil pengujian yang masih menyisakan beberapa fitur yang tidak muncul ketika analisis persyaratan bukan berarti sistem tidak bisa berjalan, sistem masih dapat berjalan memenuhi kebutuhan aktivitas pengguna sembari peneliti dapat melakukan penambahan fitur (strategi *incremental*) atau perbaikan /penyempurnaan fitur yang ada sebelumnya (strategi *iterative*).
4. Pengujian kompatibilitas dan *usability* adalah bukan tanpa alasan melainkan untuk menguji apakah peneliti telah berhasil merealisasikan kebutuhan non fungsional dari *stakeholder* yaitu sistem mudah dipakai dan dipelajari lalu mampu diakses menggunakan platform apa saja, itulah alasan mengapa peneliti menggunakan kuesioner USE, karena parameter penilaian yang dibutuhkan untuk diuji adalah penggunaan yang mudah dan mudah dipelajari, dimana 2 parameter tersebut terdapat di dalam kuesioner USE yaitu *ease to use* dan *ease to learn*. Hasil pengujian *usability* menunjukkan skor rata-rata diatas 60% yang berarti sistem berguna dan parameter *easy to use* bernilai 79% sedangkan parameter *easy to learn* bernilai 73%, hasil ini membuktikan sistem berhasil dikembangkan karena mudah dipelajari dan digunakan. Sedangkan dengan pengujian skenario

yang dirancang sesuai dengan rencana perancangan sistem membuktikan penurunan waktu yang signifikan, sehingga terbukti dengan sistem ini mampu mempercepat aktivitas mencari data dan pendaftaran dari seksi rehabilitasi BNN Kota Malang.

8.2 Saran

Saran untuk penelitian lebih lanjut pada sistem informasi data pasien rehabilitasi BNN Kota Malang, adalah sebagai berikut:

1. Penerapan sistem informasi ini beserta pemasangan infrastruktur untuk berjalannya sistem ini perlu dipertimbangkan lagi seperti manajemen risikonya jika sistem ini dipasang di BNN Kota Malang.
2. Pengujian yang lebih kompleks seperti UAT dapat menjadi pertimbangan agar sistem yang sudah dikembangkan dapat lebih tepat sasaran, dan pengguna sistem dapat menerima dan menggunakan sistem dengan maksimal karena manfaat dari sistem yang dibutuhkan pengguna dapat meningkatkan efektivitas dan produktivitas pengguna.
3. Diperlukan adanya evaluasi antarmuka pengguna, untuk meningkatkan *usability*, karena terbukti dari pengujian kompatibilitas, sistem masih belum kompatibel dengan beberapa browser dan platform sehingga itu dapat mengurangi pengalaman pengguna dalam menggunakan sistem dan juga terbukti dari pengujian *usability* yang masih dapat ditingkatkan lagi.
4. Diperlukan pengujian keamanan informasi, mengingat data-data yang akan disimpan dan diolah di sistem ini merupakan data pribadi, sehingga sangat penting untuk menjaga agar data tidak tersebar ke pihak-pihak yang tidak bersangkutan dan tidak bertanggung jawab.
5. Diperlukan adanya panduan untuk memasang sistem informasi tersebut, termasuk penyelesaian masalah ketika terjadi masalah atau hal-hal diinginkan, tidak dipungkiri mungkin masih ada kekurangan atau kesalahan dalam pengembangan sistem ini, jadi panduan untuk pemulihan dibutuhkan oleh seksi rehabilitasi BNN Kota Malang.

DAFTAR PUSTAKA

- Agarwal, B. B., Tayal, S. P., & Gupta, M. (Mahesh). (2010). *Software engineering and testing: An introduction*. Jones and Bartlett Publishers.
- Becker, J., Kugeler, M., & Rosemann, M. (2003). *Preparation of Process Modeling. Process Management: A Guide for the Design of Business Processes*.
<https://doi.org/10.1007/978-3-540-24798-2>
- Bell, D. (2003). UML basics: An introduction to the Unified Modeling Language. Retrieved February 17, 2018, from
<https://www.ibm.com/developerworks/rational/library/769.html>
- Bittner, K., & Spence, I. (2002). *Use Case Modeling*. Addison-Wesley Professional.
- Booch, G., Maksimchuk, R. a., Engle, M. W., Young, B. J., Conallen, J., & Houston, K. a. (2007). *Object-Oriented Analysis and Design with Applications. Chemistry & biodiversity* (Vol. 1). <https://doi.org/10.1145/1402521.1413138>
- Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *Unified Modeling Language User Guide, The (Addison-Wesley Object Technology Series). Elements*. Retrieved from
<http://scholar.google.com/scholar?q=intitle:The+Unified+Modeling+Language+Reference+Manual#0>
- Churm, T. (2012). An Introduction To Website Usability Testing - Usability Geek. Retrieved July 13, 2018, from <https://usabilitygeek.com/an-introduction-to-website-usability-testing/>
- Cloutier, R., & Griego, R. (2008). Applying object oriented systems engineering to complex systems. *2008 2Nd Annual Ieee Systems Conference*, 409–414.
<https://doi.org/10.1109/SYSTEMS.2008.4519058>
- Cockburn, A. (2008). Using Both Incremental and Iterative Development. *CrossTalk: The Journal of Defense Software Engineering*, 21(5), 27–30. Retrieved from
<http://alistair.cockburn.us/Using+both+incremental+and+iterative+development>
- Fowler, M. (1997). A Survey of Object Oriented Analysis and Design Methods, 653–654. <https://doi.org/10.1109/ICSE.1997.610451>
- Fowler, M. (2004). UML Distilled: A Brief Guide to the Standard Object Modeling Language. *Pearson Paravia Bruno Mondad*, 175.
<https://doi.org/10.1109/MS.2005.81>
- Grosskopf, A., Decker, G., & Weske, M. (2009). *The Process: Business Process Modeling Using BPMN*. Meghan-Kiffer Press. Retrieved from
<https://books.google.co.id/books?id=UaVZAAAAYAAJ>
- Jervis, M., & Masoodian, M. (2014). How do people attempt to integrate the management of their paper and electronic documents? *Aslib Journal of*

- Information Management*, 66(2), 134–155. <https://doi.org/10.1108/AJIM-01-2013-0007>
- Kaner, C. (2003a). An introduction to scenario testing. *Center for Software Testing Education and Research*, (June), 1–10. Retrieved from <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:An+Introduction+to+Scenario+Testing#1>
- Kaner, C. (2003b). What Is a Good Test Case? *Software Testing Analysis & Review Conference (STAR East)*, 1–16.
- Kendall, S., & Apress. (2004). *Fast Track UML 2.0*. Berkely, CA, USA: Apress.
- Leffingwell, D., & Widrig, D. (2003). *Managing Software Requirements: A Use Case Approach, Second Editio*.
- Lund, A. M. (2001). Measuring usability with the USE questionnaire. *Usability Interface*, 8(2), 3–6. <https://doi.org/10.1177/1078087402250360>
- Monk, E. F., & Bret J. Wagner. (2012). *Concepts in Enterprise Resource Planning. Statewide Agricultural Land Use Baseline 2015* (Vol. 1). <https://doi.org/10.1017/CBO9781107415324.004>
- Object Management Group (OMG). (2011). Business Process Model and Notation (BPMN) Version 2.0. *Business*, 50(January), 170. <https://doi.org/10.1007/s11576-008-0096-z>
- Peraturan Kepala BNN No. 3 Tahun 2015. (2015). Indonesia.
- Pressman, R. S. (2010). *Software Engineering A Practioner's Approach*. McGraw-Hill (Vol. 33). <https://doi.org/10.1109/6.476732>
- Przybyłek, A. (2011). Bridging the gap between business process models and use-case models. *Przybylek.Wzr.Pl*. Retrieved from http://przybylek.wzr.pl/Papers/IWORE_Przybylek.pdf
- Ramadan, J., & Sodermark, N. (2014). *Development of an Electronic Medical Record System*. Retrieved from <http://www.medscape.com/viewarticle/405802>
- Rickman, D. M., & Company, R. S. (2001). ANALYSIS AND DESIGN, 6.
- Risnita, R. (2012). Pengembangan Skala Model Likert. *EDU_BIO/ Jurnal Pendidikan Biologi*, 3.
- Sommerville, I. (2010). *Software Engineering. Software Engineering*. <https://doi.org/10.1111/j.1365-2362.2005.01463.x>
- Spool, J., & Schroeder, W. (2001). Testing web sites: Five users is nowhere near enough. *CHI '01 Extended Abstracts on Human Factors in Computing Systems - CHI '01*, 285. <https://doi.org/10.1145/634067.634236>
- Sugiyono. (2014). *Metode Penelitian Kuantitatif, Kualitatif Dan R&D*. Bandung: Alfabeta.

VisualParadigm. (n.d.). Conceptual, Logical and Physical Data Model. Retrieved March 12, 2018, from https://www.visual-paradigm.com/support/documents/vpuserguide/3563/3564/85378_conceptual,l.html

Vuksanovic, I. P., & Sudarevic, B. (2011). Use of web application frameworks in the development of small applications. *2011 Proceedings of the 34th International Convention MIPRO*, (November), 458–462.

Weske, M. (2012). *Business Process Management. Business Process Management*. <https://doi.org/10.1007/978-3-642-28616-2>

What are Conceptual, Logical and Physical Data Models? | . (n.d.). Retrieved March 12, 2018, from <http://www.datamodel.com/index.php/articles/what-are-conceptual-logical-and-physical-data-models/>

